A Secure and Efficient Method for Communications and Energy Consumption Management in IoT Wireless Sensor Networks

THESIS

submitted in accordance with the requirements for the degree of

DOCTOR OF PHILOSOPHY PROGRAM

in

COMPUTER SCIENCE

by

Safwan Mawlood Hussein Hussein, MSc

International PhD School (EIDUAL)
University of Almería
Almería, Spain
November 2023

The title of the Ph.D. thesis is in the English language

A Secure and Efficient Method for Communications and Energy Consumption Management in IoT Wireless Sensor Networks

Título de la Tesis Doctoral en Español

Método Seguro y Eficiente para la Gestión de Comunicaciones y Consumo de Energía en Redes de Sensores Inalámbricos de IoT

Candidate Full Name: Safwan Mawlood Hussein Hussein
Director: Dr. José Antonio Álvarez Bermejo
Codirector: Prof. Dr. Juan Antonio López Ramos

CERTIFICATION PAGE

We certify that we have read this thesis entitled "A Secure and Efficient Method for Communications and Energy Consumption Management in IoT Wireless Sensor Networks" by Safwan Mawlood Hussein and, as a committee, have examined the student's content and what is related to it. We approve that it meets the standards of a thesis for the degree of Doctor of Philosophy in Computer Science at the University of Almeria, Spain.

Examiners:

1. Signature:

2. Signature:

3. Signature:

Chair: Signature:

Director: Dr. José Antonio Álvarez Bermejo Signature:

Codirector: Prof. Dr. Juan Antonio López Ramos Signature:

Coordinator of the PhD Program in Computer Science:

Prof. Dr. Luis Fernando Iribarne Martinez Signature:

DECLARATION

This thesis represents the original research work conducted by me, and no part of it has been submitted elsewhere for any other degree or qualification. All the work presented in this thesis is my own, except where explicitly stated otherwise. The research was carried out under the supervision of Prof. Dr. Juan Antonio López Ramos and Dr. José Antonio Álvarez Bermejo at the International PhD School (EIDUAL) of the University of Almería.

I affirm that this work meets the standards of academic integrity and has been conducted in accordance with the ethical guidelines of the University.

Signature

Full Name: Safwan Mawlood Hussein Hussein

Date: November 2023

ACKNOWLEDGMENT

I am deeply grateful and humbled by the support and guidance provided by my exceptional supervisors, Prof. Dr. Juan Antonio López Ramos and Dr. José Antonio Álvarez Bermejo. Their expertise and knowledge have been invaluable in shaping my research, and their guidance and recommendations have been instrumental in helping me navigate the complexities of this journey. Their unwavering support and encouragement have been an inspiration, and I will forever be thankful for their invaluable contributions to my academic and personal growth.

I also extend my heartfelt thanks to my lovely family and friends, who have provided unwavering emotional support and love. Their encouragement and understanding have been a source of strength, and I could not have completed this journey without their support.

I am deeply grateful for the opportunities and support provided by my supervisors, and I will always remember their contributions to my success.

DEDICATIONS

This thesis is a gift to my mother's soul for all the love, support, and encouragement she has given me over the years.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
API	Application Programming Interface
ATM	Automated Teller Machine
BD	Burmester and Desmedt Protocol
BLE	Bluetooth Low Energy
BS	Base Station
BSS	Basic Service Set
BW	Bandwidth
CDMA	Code Division Multiple Access
CKA	Conference Key Agreement
CoAP	Constrained Application Protocol
CPS	Cyber-Physical System
CPU	Central Processing Unit
DFM	Diffie Hellman for Multicast
D-FT	Distributed Flat Table
DHT	Distributed Hash Table
DLT	Decentralised Ledger Technologies
D-OFT	Distributed One-way Function Tree
D-LKH	Distributed Logical Key Hierarchy
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECEG	Elliptic Curve El Gamal
ECIES	Elliptic Curve Integrated Encryption Scheme
EdDSA	Edwards-curve Digital Signature Algorithm
EHGUC_OAPR routing	Energy harvesting-based clustering and optimal adaptive performance
FHMQV	Full-Hierarchy MQV
FS	Full System.

GKM	Group Key Management
GPS	Global Positioning System
HTTP	HTTP: Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
IPFS	Interplanetary File System
ISA	Instruction Set Architecture
ITU	International Telecommunication Union
JSON	JavaScript Object Notation
LPWAN	Low-Power Wide Area Network
LTE	Long-Term Evolution
M2M	Machine to Machine
MAC	Medium Access Control Microcontroller
MQTT	Message Queuing Telemetry Transport
MVD	Minimum Viable Device
NFC	Near Field Communication
OAPR	Optimal Adaptive Performance Routing Algorithm
OSI	Open Systems Interconnection
OWASP	Open Web Application Security Project
POS	Point-of-Sale terminal
QoS	Quality of Service
RAM	Random Access Memory
SCADA	Supervisory Control and Data Acquisition
SDOs	Standard -Developing Organizations
REST	Representational State Transfer
RFC	Request for Comments
RPL	Routing Protocol for Low-Power and Lossy Networks
RTOS	Real Time Operating System
STR	Skinny Tree Protocol
TDMA	Time-Division Multiple Access
TP	Transmission Power
VLSI	Very large-scale integration

WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WSN	Wireless Sensor Network

RESUMEN

Las redes inalámbricas de sensores se están convirtiendo en un componente importante del Internet de las cosas. Consiste en sensores limitados en términos de recursos, energía y procesamiento, varios tipos de controles y nodos de puerta de enlace. WSN proporciona soluciones variantes para muchas aplicaciones, incluidas la atención de la salud, la agricultura, el medio ambiente, la industria, la defensa y muchos otros campos. Debido al escenario de implementación y los métodos de comunicación utilizados en tales redes, se requiere un protocolo sólido y seguro que incluya medidas para permitir que solo dispositivos autenticados y aprobados se unan a la red. Además, los datos que se transmiten en esta capa de la red deben validarse y protegerse contra escuchas y alteraciones. La aplicación de métodos de seguridad tradicionales a un sistema IoT y WSN es un desafío debido a su topología descentralizada y los recursos limitados de estos dispositivos.

El uso de métodos criptográficos para cifrar datos y configurar canales seguros para comunicaciones de protocolo es una parte clave de la seguridad de IoT y WSN. Los dispositivos de sensores e IoT suelen tener menos recursos debido a su tamaño y naturaleza. Esto tiene el efecto de evitar que la mayoría de los dispositivos de sensores e IoT tengan la potencia de procesamiento o los recursos necesarios para las técnicas de cifrado más potentes. Podrían utilizarse técnicas ligeras de encriptación porque todavía se requiere encriptación para su funcionamiento.

En este trabajo, exploramos los avances más recientes en la gestión segura y eficiente de claves de mensajes de multidifusión, así como los nuevos desafíos que presentan estos nuevos enfoques. Nos concentramos en la gestión de claves de grupo en WSN además de los protocolos de enrutamiento. Mediante experimentos prácticos, evaluamos las propiedades de algunos protocolos de administración de claves grupales, como el acuerdo clave para los mensajes transmitidos y recibidos, el tiempo de cómputo, el uso de la memoria y los números redondos de recodificación y la confiabilidad del protocolo. Al utilizar estas propiedades, propusimos un protocolo eficiente y liviano de administración de claves de grupo que utiliza criptografía de curva elíptica para garantizar la seguridad de la comunicación de los nodos y un protocolo de

enrutamiento mejorado basado en el protocolo LEACH para demostrar un mejor rendimiento en parámetros como la vida útil de la red, los nodos muertos, y consumo de energía.

Demostramos que el método propuesto es mucho más receptivo, altamente escalable y eficiente en energía, reduce el tiempo de cómputo y la cantidad de rondas para iniciar claves y usa menos memoria.

En general, este trabajo presenta evidencia teórica y empírica de que el método propuesto es capaz de abordar los problemas de seguridad de las WSN y aumentar la vida útil de la red.

ABSTRACT

Wireless sensor networks (WSN) are becoming an essential component of the Internet of Things (IoT). It consists of constrained sensors in terms of resources, power, and processing, various types of controls, and gateway nodes. WSN provides variant solutions for many applications, including health care, agriculture, the environment, industry, defense, and many other fields. Due to the deployment scenario and communication methods used in such networks, a robust and secure protocol is required that includes measures to only allow authenticated, approved devices to join the network. In addition, the data that is transmitted at this layer of the network must be validated and protected from eavesdropping and alteration. Applying traditional security methods to an IoT and WSN system is challenging due to its decentralized topology and the constrained resources of these devices.

The use of cryptographic methods to encrypt data and set up secure channels for protocol communications is a key part of IoT and WSN security. IoT and sensor devices typically have limited resources because of their size and nature. This has the effect of preventing the majority of IoT and sensor devices from having the processing power or resources required for the more powerful encryption techniques. Lightweight encryption techniques could be utilized because encryption is still required for their operation.

In this work, we explore the most recent advancements in secure and efficient key management of multicast messages as well as the new challenges introduced by these new approaches. We concentrate on group key management in WSN in addition to routing protocols. Using practical experiments, we evaluate the properties of some group key management protocols, such as key agreement for transmitted and received messages, computational time, memory usage, round numbers of rekeying, and protocol reliability. Utilizing these properties, we proposed an efficient and lightweight protocol for group key management using elliptic curve cryptography(ECC) method to ensure the security of node communication. Additionally, we have devised an enhanced routing protocol, building upon the LEACH protocol, which exhibits significant improvements in crucial parameters such as network lifetime, dead nodes, and energy consumption.

We demonstrate that the proposed method is much more responsive, highly scalable, and energy efficient; it reduces computational time and the number of rounds to initiate keys; and it uses less memory.

Overall, this work presents theoretical and empirical evidence that the proposed method is capable of addressing WSN security challenges and increasing the network's lifetime.

CHAPTER 1
INTRODUCTION

1.1 Background

The United Nations predicts that the world's population will continue to grow significantly, reaching 8.5 billion people by 2030, 9.7 billion by 2050, and 11.2 billion by 2100, with the majority of this growth taking place in urban areas. This rapid urbanization has created substantial pressure on cities' infrastructure and services to support population growth while still prioritizing sustainability and environmental goals. The growth of cities has also created excessive burdens on resources such as food, energy, the environment, and lifestyles. In order to meet the food demands of this expanding population, the agriculture industry will need to adopt new technologies to efficiently manage existing resources [1]. Technologies play a vital role in sustaining economic progress, and the technology likely to have the greatest impact in the next few years is the IoT. Integration of IoT devices enables the agriculture sector to increase productivity, which has occurred at a lower cost with efficient use of resources.

Over the next few years, the use of intelligent solutions made possible by the internet of things will increase in agricultural operations. In fact, just a handful of the most recent reports state that the installation of Internet of Things devices in the agricultural sector will have a compound annual growth rate of 20%. Additionally, It is anticipated that by 2024, there will be 225 million connected devices being utilized in agriculture, up from 13 million in 2014 [2].

Researchers have proposed a variety of solutions to address the problems in the agriculture sector, mentioning the impact of IoT technologies on product quality and quantity as well as the effective use of resources in the field. For example, the authors of [3] discuss the potential effects that climate change could have on resources, including a shortage of available water, an increase in the amount of soil salinity, and the irrigation that is required. The work shows the positive impact of using IoT devices on water management during irrigation processes.

IoT-based devices have emerged as a promising tool for the agricultural sector, enabling the collection of real-time data from a variety of sensors that monitor critical environmental parameters such as soil moisture, air quality, and temperature. These devices are being used for various purposes, including weather monitoring, irrigation, grazing management, and crop and livestock monitoring, to optimize resource usage, reduce waste, and improve productivity.

The data collected by IoT sensors can be analyzed and visualized on an agricultural dashboard, allowing farmers to interpret and act upon the data in real-time. The dashboard provides a customizable interface for presenting data on crop and livestock conditions, environmental factors, and other parameters, enabling farmers to make informed decisions about their operations.

Wireless sensor networks (WSNs) are regarded as a key building block of IoT technologies. It is comprised of a limited number of sensor nodes that can sense or regulate physical characteristics such as sound, light, temperature, humidity, and others in a geographical area. Sensor nodes have constrained energy, memory, and CPU capabilities. Node components are a power unit, a processing unit, one or more sensing units, a transceiver, an antenna, and optional components like a position-finding system, a power generator, and an actuator. The volume of sensor nodes varies from cubic nanometers to cubic decimeters.

WSNs are used in many applications, including medical care [4], tracking, environmental monitoring, building automation, the military, and precision agriculture [5-7], environmental condition monitoring [8, 9], control of machines and processes [10, 11], automatization of monitored areas, and monitoring systems [12].

In a system of this complexity, which may include a huge number of low-cost sensor nodes, it is essential to consider the factors that affect data communication and data integrity [13]. Due to the limited availability of resources such as memory, energy, computation, and communication, wireless sensor networks are desirable targets for various attacks [14]. Hence, to reduce the risk of security breaches, it is important to strike a balance between the resources required to implement security measures and the potential impact on business operations. Secure node-group communication will ensure the authenticity, confidentiality (including forward and backward secrecy when nodes leave or join the network), and integrity of the messages being sent and received [4].

Group key management is widely used to secure a variety of applications, including IoT systems based on WSNs [14]. The advantages of using group key management protocols are low energy consumption, memory usage, and communication overhead, especially for networks with high scalability.

In recent years, many studies have been conducted on group key agreements, and various solutions have been proposed by scholars based on different cryptosystems to achieve security requirements in WSN. However, key management and distribution remain a challenge due to the nature of these networks. Design group key establishment protocols become more complicated in such networks, especially when the size of the network groups changes frequently and the updating process reduces the scheme's effectiveness and scalability [15, 16].

Whitfield Diffie and Martin Hellman (DH) introduced the first protocol of public key cryptography using private and public keys [17]. The protocol was designed to exchange keys over open channels between two parties only. Scalability and security breaches are the main issues with the DH protocol; hence, multiple strategies have been presented for reducing the size of the key in secure group communication using distributed protocols. The majority of these methods are based on variants of the n-parity Diffie-Hellman key agreement protocol.

The primary issue with such systems is that the size of the asymmetric key is larger due to the fact that network overhead has increased [17]. While [18] proposed the Group Diffie-Hellman key exchange (GDH) protocol to make the previous protocol available for a group of members, Cliques [19] introduces a new method for secure group key agreement. It can handle a large number of participants and provides stronger security guarantees, such as resilience against active attacks. In addition, the authors claim that their protocol is more scalable and efficient than previous group key agreement protocols, including those based on DH.

There are many authentication protocols and key agreement methods that have been developed over the years. Each of these protocols has its own strengths and weaknesses and is designed for specific use cases and security requirements. Other researchers introduced authentication protocols and key agreement methods. Shin, S., and Kwon [20] proposed a simple authentication method for WSN based on the three-factor approach and key agreement protocol.

The elliptic curve scheme is a well-known lightweight public key cryptography that is found to be efficient for computing and secure for low-power devices [18, 21]. For instance, references [22] use Elliptic Curve Cryptography (ECC)-based implicit certificates and the Elliptic Curve Diffie-Hellmann (ECDH) method to establish a secure key for unicast communication in WSNs and IoT. In contrast, [23] has presented a review paper discussing various types of group key establishment methods, such as RSA and elliptic curves. The paper shows that those methods require complex computations and high storage space, which may not be appropriate for applications that rely on resource-constrained devices [24].

In WSN systems, the physical layout of the network should be carefully designed to avoid any kind of loss of data. Sensor nodes will gather data, which they will then promptly transmit to the base station or trigger in response to events [25, 26]. Due to the constraints, it is required to have an appropriate protocol at the MAC level in order to enable efficient use of energy inside the network, such as sleep and wake-up patterns, which are known as nodes' duty cycles, to increase the network's lifetime. In addition, the high density of nodes in a network forces the use of routing protocols that use energy efficiently to transmit or receive data to other nodes or to the central point or base station.

Researchers have proposed a variety of algorithms and protocols. Their primary objective is to optimize energy consumption and extend network lifetime, as battery replacement can be difficult in certain applications and node failure can be costly [27]. It has been proven that clustering is an energy-efficient and scalable approach [28]. In recent years, numerous cluster-based routing techniques have been suggested for WSNs. Although many of them built some sort of energy-efficient cluster, few of them properly analyzed the target application situations, for instance, when constructing clusters, one must consider the impacts of diverse physical propagation mechanisms in the environment.

The LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol is a hierarchical clustering-based protocol specifically designed for wireless sensor networks (WSNs). Its primary objective is to minimize energy consumption and prolong the overall lifetime of the network. Heinzelman, Chandrakasan, and Balakrishnan first introduced the protocol [29]. Since its inception, LEACH has received a lot of attention and interest from the research community, mainly due to its notable energy efficiency, simplicity, and load-balancing capabilities. LEACH is a cluster-based routing and MAC-layer approach that incorporates application-

specific data aggregation to reduce energy consumption. In order to prevent excessive energy consumption from a single node, the cluster head role is rotated among the nodes. The protocol achieves energy efficiency by evenly distributing the energy load among the nodes and allowing each node to have an equal chance of becoming a cluster head.

The protocol has been widely studied and evaluated for its performance in various scenarios. Numerous studies have shown that LEACH provides significant improvements in energy efficiency compared to non-clustered-based routing approaches, making it a benchmark protocol in the field [30]. In addition, several variations of the LEACH protocol have been proposed to improve its performance in specific applications. For instance, the LEACH-C protocol is a centralized version of LEACH in which the base station, which receives all sensor node information after each round, performs cluster creation and cluster head selection [31].

The protocols mentioned above are known to have a high number of transmitted messages, resulting in the need for many rounds of processing during the initial key agreement. This poses a challenge for key management for IoT and WSNs. To address this challenge, a proposed protocol based on public-key cryptography using elliptic curves has been introduced. This protocol has the potential to solve key management issues in IoT and WSNs by reducing the number of transmitted messages and processing rounds required during the initial key agreement. Therefore, it presents a promising solution for the issues previously mentioned.

1.2 Motivation

In In recent years, the IoT has gained popularity and grown remarkably. Wireless sensor networks (WSNs) are a key building block of IoT technologies in use in many applications. Providing efficient security services in wireless sensor networks is an active area that presents numerous challenges and research opportunities. A lack of security and privacy-related problems are solved using common cryptographic techniques. However, cryptographic protocols are resource-intensive, and sensors are resource-constrained. With these issues in mind and motivated, we proposed a lightweight protocol to secure WSN data sharing.

Three major findings from recent research on security issues in WSNs and energy-efficient routing protocols have fueled our motivation. First and foremost, reliability is an essential feature of any network protocol. Before delivering every data packet, most algorithms attempt to discover the best route from source to destination. This technique is prone to faulty transmission when intermediate nodes change their configuration (power, location, transmission mode, etc.) on a frequent basis. As a result, the need for a protocol that can ensure transmission reliability at each forwarding node arises.

Second, when faulty sensor nodes appear, it can have a dramatic effect on the performance of the routing protocol. This occurs when sensors become decalibrated or experience adverse weather conditions, leading to incorrect behavior. In response, protocols strive to find alternative paths from the source or malfunctioning nodes to the destination. The objective is to retransmit the data with minimal disruption. Addressing the challenges posed by these faulty nodes is essential for maintaining effective and efficient communication within the network. By identifying alternate paths and adapting to the presence of faulty nodes, the routing protocol aims to ensure reliable data transmission despite the potential disruptions caused by sensor malfunctions or environmental factors.

1.3 Security in Wireless Sensor Networks

Wireless Sensor Networks (WSNs) frequently lack centralized control and are frequently utilized in challenging conditions with unreliable communication connections. As a result, depending solely on typical IT network solutions to mitigate security in WSNs is insufficient. The complexity of WSN networks, type of devices, protocols, and routing systems, combined with the availability of various services, exacerbates the problem. Hence, existing security techniques are insufficient to provide comprehensive protection.

Various strategies and trends have evolved to address this issue, with the goal of achieving specific security levels within wireless sensor networks. These approaches include trust management and lightweight encryption technologies, which enable low-power and resource-constrained devices with cost-effective encryption solutions. Despite these advancements, routing protocols are susceptible to significant attacks, including the injection of false or malicious

routing information. Consequently, such attacks might cause delays and packet losses attributable to routing conflicts.

Numerous strategies, including the utilization of encryption and multiple node information correlation, have been proposed to mitigate routing attacks in WSNs [32]. When formulating a protocol, it is imperative to achieve a harmonious balance between two divergent aims: ensuring security and optimizing performance. Achieving good performance at security levels may require trade-offs in terms of low energy consumption, processing requirements, and storage usage, and vice versa. The implementation of key management strategies has emerged as a viable approach to ensuring the security of wireless sensor networks [33].

Based on security policy, key management is a comprehensive framework comprising activities and mechanisms that facilitate key distribution and ensure compliance with the keying process criteria among nodes. It encompasses key generation, maintenance, distribution, protection, and control to ensure the secure and appropriate utilization of cryptographic keys. Considering the ability to update keys in sensor nodes, key management techniques are classified as static or dynamic [34]. Static key management involves pre-distributing a fixed set of keys to sensor nodes prior to network deployment. These keys remain constant throughout the network's lifespan and are utilized for message encryption and decryption. While static key management offers simplicity and efficiency, it presents security vulnerabilities such as susceptibility to attacks and compromised keys. In contrast, dynamic key management employs dynamically generated keys that are periodically distributed and updated during network operation. The objective of dynamic key management algorithms is to enhance network security by reducing key exposure time and mitigating the risk of compromised keys. By refreshing keys at regular intervals, dynamic key management enhances the resilience and integrity of the cryptographic system within the network.

Group key management is broadly used in various modern collaborative and distributed applications to provide security. According to source [34], group key management has been identified as a potentially more efficient approach compared to individual key management in terms of energy consumption and memory usage, particularly in large-scale sensor networks. On the other hand, paper [25] explores several techniques aimed at optimizing key management in IoT devices. These techniques include the utilization of lightweight public key cryptography algorithms, the implementation of threshold-based key sharing mechanisms, and the involvement

of trusted third parties to offload key management tasks. These approaches contribute to enhancing the overall efficiency and security of key management in IoT environments.

1.4 Objectives

The main objective of this study is to address the challenges of secure communication in a Wireless Sensor Network (WSN) environment. The focus is on developing efficient group key agreement protocol that able to enhance various aspects of the system, such as reducing computational time, memory usage, power consumption, and the number of communication messages required for key establishment and rekeying. Furthermore, the goal is to achieve excellent network responsiveness and scalability.

The work intends to create a secure foundation for communication within the WSN by adopting efficient group key agreements. The goal is to optimize these agreements in order to reduce resource use while maintaining a high level of security.

Furthermore, the thesis aims to address challenges related to network lifetime, energy consumption, and load balancing at the cluster head. An efficient routing protocol will be developed to achieve these goals. This protocol will take into account variables such as energy efficiency, load balancing, and network scalability to ensure the WSN operates optimally.

The project intends to increase the operational length of the WSN without requiring frequent battery replacements or recharging by extending network lifetime and lowering energy consumption. This helps to save maintenance expenses and efforts. Furthermore, load balancing at the cluster head will aid in the distribution of network traffic across nodes, preventing congestion and enhancing overall network performance.

Overall, the objective is to design and implement a secure and efficient WSN environment by focusing on group key agreements and an efficient routing protocol. This work aims to enhance the security, responsiveness, scalability, energy efficiency, and load balancing capabilities of the network, thereby enabling the successful deployment of WSNs in various real-world applications.

1.5 Main Objectives

How could authentication and data integrity be managed to ensure the security and privacy of the IoT-based WSN? To provide a full answer to this question, we propose a research method that integrates theoretical analysis with empirical research and is motivated by the sub-questions listed below:

R1) What are the challenges in securing wireless sensor networks using efficient group key management?

R2) How can we secure the management of sensors in large-scale deployments?

R3) How does the proposed method reduce the cost of transmission and power consumption in wireless sensor networks?

R4) Which routing protocol is sufficient to optimize energy consumption and prolong the network's lifetime?

1.6 Significance of the Research

This work proposes an efficient and lightweight method of group key management using elliptic curve cryptography to solve the security issues of devices with constrained resources, such as IoT-based WSN. The proposed protocol is highly scalable, consumes less power, reduces computational time and the number of rounds to initiate keys, and uses less memory for group key agreement. Furthermore, the work extends the network's lifetime by enhancing one of the most widely recognized routing protocols, LEACH.

1.7 Organization of this Thesis

The present chapter presents the introduction of the research in general, the background of the problem statement, and why the project was carried out. It also talks about the research questions and goals, as well as the study's scope and importance. The context of this thesis is presented in Chapter 2, which provides background information and related work. We investigate and discuss the characteristics of IoT and wireless sensor networks, paying special attention to

the security issues in WSN and the security measurements that provide a secure environment for communication in an open area. It also gives an overview of group communication methods, group key management, and routing protocols. It also talks about the problems and limitations of group key management and the routing methods and techniques used in previous research. Chapter 3 explains how the project will be done by listing the materials, methods, and technology that will be used to get the information needed for this research to be successful. In Chapter 4, we investigate the scalability of the proposed method. We develop a comprehensive link model of the proposed method and a simulator to analyze its performance under various conditions. Analysis shows that the proposed protocol consumes less time and power and increases the lifetime of the network. Chapter 5 presents the conclusion and recommendation for the future work of this thesis.

CHAPTER 2
LITERATURE REVIEW

This chapter serves to introduce the fundamental concepts and technologies explored in this thesis. It provides a concise overview of authentication and key establishment, group key establishment, and the different types and roles of group key management. Furthermore, we briefly discuss routing protocols in WSN, with a particular emphasis on network structure and hierarchical routing, such as LEACH (Low-Energy Adaptive Clustering Hierarchy), an energy-efficient communication protocol for wireless microsensor networks.

The chapter thoroughly examines the security issues surrounding WSN, addressing the challenge of achieving secure communication over open channels among a group of sensors using lightweight cryptography encryption. Various approaches for encrypting data are explored, and the methods used to enhance the efficiency of WSN are discussed. Lastly, the chapter sheds light on diverse strategies that can be implemented to bolster the security and extend the network lifetime of WSN.

2.1 Introduction to Authentication and Key Establishment

The establishment of secure communications is a fundamental process that encompasses authentication and key establishment as its foundational components. Key establishment focuses on acquiring robust cryptographic keys to safeguard the integrity and confidentiality of the transmitted data. On the other hand, authentication pertains to verifying the identities of the involved parties in the communication process. The escalating reliance of contemporary society on digital networks accentuates the paramount importance of ensuring the security of communication systems. As our world advances and incorporates technological innovations, the protection of these systems assumes an indispensable role in facilitating the seamless operation of our interconnected society. Moreover, it is anticipated that the significance of ensuring the safety of communication systems will intensify further in the foreseeable future [35].

2.1.1 Group Key Management

The group key management scheme is broadly classified into two main categories: dependent and independent network-based. Each of the mentioned categories can be further classified. For instance, network-independent key management protocols are classified into centralized, decentralized, and distributed key management protocols, while network-dependent protocols are classified as tree-based and cluster-based key management [36].

Centralized group key management protocols:

One member of the group is superior to the other members and is assigned to control access by generating and distributing keys. The main challenges in this category are a reduction in storage and power consumption during the computational process. The Key Distribution Center (KDC) is available to assign a secret key to each member of the group.

Decentralized group key management protocols:

It is using the method of sharing authority by dividing a large group's multicast communication into subgroups to reduce the issue of concentrating on one single place for generating and distributing keys.

Distributed key management protocols:

Distributed key management protocols are used in distributed systems to manage cryptographic keys securely across different nodes or participants. These protocols ensure that keys are distributed and shared in a secure and reliable manner so that only authorized entities have access to the keys. distribution center [37].

2.2 Key Management Role

The main role of a key management scheme is to apply access control to the key in group communication. It provides support for establishing, processing, and protecting keys among members of a specific group according to restrictions specified by the group [37, 38]. A key

management scheme provides authentication for a communication channel. When validating user identity is necessary to access a set of resources in group communication, authenticity is required.

Hence, authentication plays a vital role in the key management scheme to prevent an intruder from mimicking a valid group member. There are many methods and protocols for authentication. A new group member cannot send or receive data inside the group unless it has been authorized by performing access control. Generating keys for group members is another duty of the key management scheme, in addition to securely distributing them. The key needs to be updated from time to time according to the group policy for security reasons [39]. Backward and forward secrecy is required for every change in group member to protect previous and new conversation communication from a new or left-over member. Independence should be considered while generating a new set of keys to avoid any kind of prediction by intruders [40].

Depending on the ability to update the cryptographic keys of sensor nodes during their run time (rekeying), these schemes can be classified into two different categories: static key management involves pre-distributing fixed cryptographic keys to sensor nodes for the entire duration of the network. While this approach simplifies key distribution and storage, it also increases the vulnerability of the network over time. As a key is used for an extended period, the probability of it being attacked rises significantly, posing a risk to the network's security.

On the other hand, dynamic key management refreshes cryptographic keys at regular intervals during the network's lifetime. By periodically replacing the keys, the network maintains a higher level of security and reduces the window of opportunity for attackers. Dynamic key management schemes provide improved resilience and security, mitigating the risks associated with using the same key for an extended period. Although they require additional computational resources and communication overhead, the benefits of enhanced security justify these costs, especially in long-lived networks or hostile environments.

Dynamic key management is regarded as promising for key management in sensor networks. In a dynamic key management process, new keys are generated and distributed to the members of the group, while the old keys are revoked or deleted. The process of key distribution and revocation needs to be done securely and efficiently to ensure that only authorized members have access to the multicast data. It can be considered a set of processes used to perform rekeying

either periodically or on demand as needed by the network. Since the keys of compromised nodes are revoked in the rekeying process, dynamic key management schemes enhance network survivability and network resilience dramatically [41].

Figure 2.1 illustrates a taxonomy of group key management schemes, adapted from the paper [42]. The taxonomy is organized into three categories: centralized, decentralized, and distributed. In the centralized approach, pairwise keys and hierarchical tree methods are used for group key management. Hierarchical tree methods are further divided into server-driven rekeying and user-driven rekeying. The decentralized approach includes two methods: independent TEK per subgroup and common TEK. With independent TEK per subgroup, each subgroup has its own unique key, while common TEK uses a single key for all subgroups.

In the distributed approach, three methods are included: ring-based operation, hierarchy-based operation, and broadcast-based operation. Ring-based operation involves transmitting keys along a circular path among group members, while hierarchy-based operation involves a hierarchical structure where keys are distributed based on the level of the hierarchy. Broadcast-based operations involve transmitting keys to all group members simultaneously. Dynamic key management schemes that employ these different methods can dramatically enhance network survivability and resilience, making them a valuable tool for securing multicast networks.

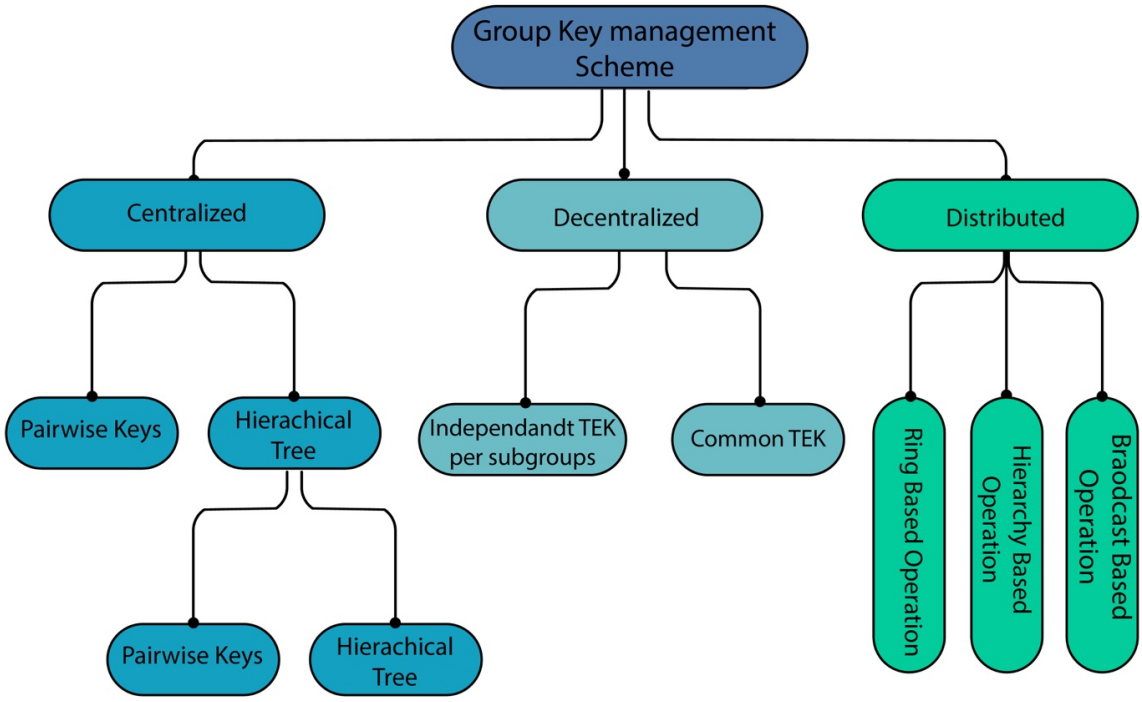

Fig. 2. 1: Taxonomy of Group Key Management Schemes [42].

2.3 Basic Requirements and Evaluation Metrics

The group key management approach provides basic security requirements for confidentiality, integrity, authenticity, and access control. The same method holds for a distributed key management scheme. Hence, the basic requirements and evaluation metrics for an efficient distributed key management scheme rely on networks and the application environment, considering some metrics such as security, flexibility, and efficiency. Some of the following metrics can be used only in a dynamic key management scheme in which keys are refreshed throughout the lifetime of the network, such as forward secrecy, backward secrecy, collusion freedom, and key connectivity, while the remaining metrics can be implemented for both dynamic and static key management schemes.

2.3.1 Security Requirement

A key management scheme must provide a cryptographic key in a secure way that prevents malicious nodes and new members or left members from encrypting and decrypting previously and future messages in group communication. For instance, the forward secrecy method is applied to prevent users who left the group from having access to any future key (for the purpose of decrypting future messages). While backward secrecy, on the other hand, operates to prevent new users who join the group recently from accessing previous keys that help decrypt previous messages. To ensure both forward and backward secrecy, a re-key operation with a new traffic encryption key is required.

Another metric that can be found under the category of security requirements is collusion resistance. Evicted members in the group should not be able to collude and share system keys, to prevent deducing the current traffic encryption key. A good dynamic key establishment technique must resist the collusion of newly joined and compromised nodes.

2.3.2 Efficiency Metrics

Any system that uses group key management protocols must be able to provide the type of keys that are needed for the exact operation of the network while, at the same time, considering the heavy constraints inherent to the network. Hence, the process of key generation and distribution should not create excessive burdens on bandwidth, processing time, memory, and energy consumption during the process of generating and distributing keys [43].

- The number of iterations required to generate and distribute keys is one of the factors that influence processing and communication requirements.
- As the number of messages within a group increases, both the number of transmitted and received messages also grow linearly, leading to elevated latency. Consequently, it is crucial for the protocol to minimize the message count, ensuring efficient communication.
- In a key management scheme, resources need to be reserved during the group setup phase due to computational requirements. More resources are required for maintaining a group

than for updating it, and during the maintenance process, communication with all members is necessary.

- DH keys identify whether the protocol uses Diffie-Hellman (DH) [36] to generate the keys. The use of DH to generate the group key implies that the group key is generated in a contributory fashion. A DH key, also known as a Diffie-Hellman key, is a shared secret key that is generated by two parties using the Diffie-Hellman key exchange algorithm. The Diffie-Hellman key is used to encrypt and decrypt messages transmitted between the two parties via an insecure communication channel.

Bandwidth: The process of re-keying generates a variable number of messages, the exact quantity depending on the method used, network type, and environment. As a result, the protocol should be designed to minimize the quantity of messages and bandwidth overhead, especially in dynamic environments.

1-affects-n: In the contect of group key management protocols the term "1 affects n" refers to an issue in which a single membership changes in the group impacts all of the other members in the group. This issue arises when a protocol requires that each and every member of a group must make a commitment to a new temporary encryption key (TEK) whenever a single membership change occurs. This could result a significant overhead and delay, especially for large groups [42].

Delay: In certain multimedia applications, an uneven delay in packet delivery is deemed unacceptable due to its detrimental effect on the overall quality. Therefore, it is essential for any key management scheme to carefully consider this aspect and aim to minimize the influence of key management on packet delivery delays. By reducing the impact of key management operations on packet delivery delays, the overall quality and performance of multicast-based applications can be significantly improved [42].

Key connectivity: It is a measure of how well groups of sensor nodes are able to establish shared keys, which is crucial for ensuring secure communication between nodes in a WSN. Local connectivity and global connectivity are two metrics used to evaluate key connectivity in WSN. Local connectivity refers to the probability that a pair of neighboring nodes can establish a shared key, while global connectivity considers the probability that groups of sensor nodes throughout the network can establish shared keys. Insufficient key connectivity can seriously

impair the functioning of the network, as neighbouring nodes will not be able to communicate securely. This can lead to the loss of data or even the complete failure of the network [44].

2.4 Distributed Key Management

In a distributed key management system, no member of the group is superior to another member of the group, and no member is assigned control of the group in terms of generating and distributing keys. No member has more authority than other members; when there is a change in the size of a membership group, all members follow a procedure defined by a group key management protocol to make a computation of the group key and distribute the result of the computation to multicast group communication through sending overhead messages.

A distributed key management scheme provides the property of fault tolerance, allowing operations to continue properly even in the event of failures. This enhances system performance, increases reliability, and reduces bottlenecks in the network compared to other approaches. However, it is important to note that a distributed key management scheme also introduces certain drawbacks. Specifically, the scheme is associated with increased communication overhead and computational time, which scale linearly with the number of members in a group.

Moreover, contributory protocols necessitate the awareness of each group member regarding the list of participants to ensure the robustness of the protocols [45]. Various parameters can affect a distributed key management approach; these include the number of iterations required by a protocol to complete the computation process, communication requirements, the number of messages transmitted and received among group members, as well as the processing and computational time needed to establish contact with all members of a group. These attributes aid in evaluating the efficiency of each distributed key-agreement protocol.

Distributed key management protocols can be classified into three sub-categories based on the type of logical topology shaped by members in a group for cooperation: ring-based cooperation, broadcast-based cooperation, and hierarchy-based cooperation, as shown in Figure 2.2. There are some factors that affected the distributed key management approach, such as the

number of iterations and messages transferred among members, in addition to the cost of generating and computing the group key [42].

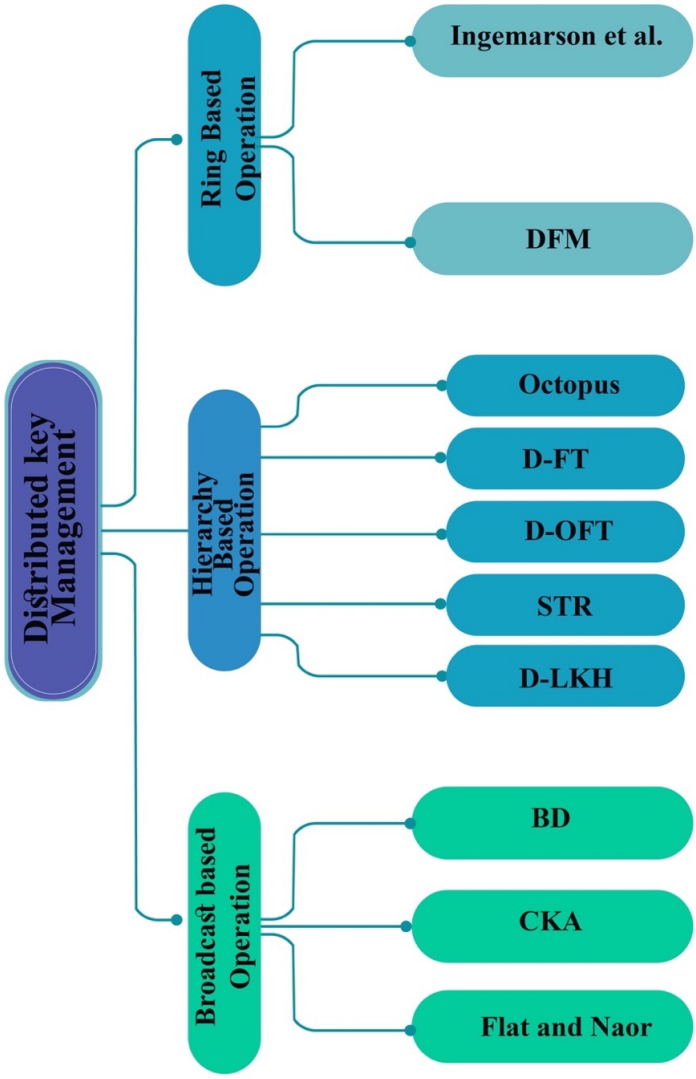

Fig. 2. 2: Distributed Key Management Schemes Categorized by Their Operation Method [42].

This diagram illustrates a taxonomy of distributed key management systems for secure group communication. The taxonomy is founded on the operation methods used for key management: ring-based, hierarchy-based, and broadcast-based. There are two methods of ring-based operation: Ingemarson et al. and DFM. Ingemarson et al. use a ring structure to distribute keys, whereas DFM employs a dynamic fault model to generate and distribute keys. D-FT, D-OFT, STR, and D-LKH are techniques of Hierarchy-based operation. Octopus is a tree-based key distribution method that employs a threshold cryptography scheme. D-FT and D-OFT utilize fault-tolerant techniques to distribute keys. STR distributes keys using a star topology, whereas D-LKH employs a modified version of the LKH algorithm. Three methods are utilized in broadcast-based operations: BD, CKA, and Flat. CKA uses a key agreement protocol to distribute keys, whereas BD employs a broadcast encryption scheme. Flat employs a tree-based strategy for key distribution. This taxonomy provides a comprehensive overview of distributed key management schemes for secure group communication and can help in selecting an appropriate method based on the specific needs and characteristics of the group.

The majority of distributed group key management protocols are derived from the cryptography Diffie-Hellman method and have attempted to extend it from a two-party setting to a multiparty environment. Steer et al. [46], for example, proposed a method that provides a high quality and robust operation that can be applied on digital networks such as the Integrated Services Digital Network (ISDN), with the encryption process managed by a conference control unit.

The system's limitations include that key distribution and authentication require a large number of transmitting messages from the control unit ($4n$-2) that must be repeated each time a new member is added to the group communication. Furthermore, this approach lacks backward and forward secrecy, which is essential for every change in-group member to protect previous and new conversation communication from newly joined or left members. In this setup, the attacker can use passive eavesdropping to intercept information being communicated over the channel.

Rafaelli and Hutchison's paper [37] presents an overview of several prominent group key management protocols, including Group Diffie-Hellman protocol, the Key Tree scheme, and the Logical Key Hierarchy scheme. In addition, the authors present a novel conference key distribution mechanism that overcomes passive eavesdropping attacks. The approach employs a two-degree cyclic function that has been proven to be secure against passive adversaries. The method is extended to be secure against active eavesdropping attacks by providing a conference key that is a combination of a public key and an authentication scheme. The authors also extend the method to be secure against an adaptively chosen message attack by a real-time middleperson, provided that the discrete logarithm problem is intractable. One potential limitation of this approach is its vulnerability to directed, chosen, and known message attacks, which could compromise the security of the system.

In 2006, [47] introduced a flexible framework for constructing group key agreement protocols that offer security against malicious insiders and active adversaries within a point-to-point network. The obtained results show that the proposed framework maintains a consistent number of rounds throughout the construction process, resulting in an effective, adaptable, and secure protocol.

In 2007, Jonathan Katz and Moti [48] addressed the problem of authenticated group key exchange among multiple parties. The authors present a provably secure protocol based on the decisional Diffie-Hellman (DDH) assumption that uses the same security model as other recent

work in the area but with improved rigor. The protocol is designed to ensure that parties involved in the key exchange are who they claim to be and to prevent impersonation attacks.

The paper [49] presents a modular approach to the design and analysis of authentication and key exchange protocols. The authors contend that the design and analysis of these protocols should take a modular approach, concentrating on the distinct security objectives that each protocol component must meet. This study gives a formal model for this technique and illustrates its effectiveness through the analysis of several existing authentication and key exchange protocols.

The paper [50] proposes a (t, n) threshold multi-conference key agreement protocol that relies on a public-key infrastructure. The protocol can establish a session key even in the presence of a number of malicious participants among the conference participants. The protocol is provably secure against passive and active adversaries under the assumption that the number of corrupted servers in the scheme is less than a certain threshold value. The paper provides a detailed analysis of the proposed protocol's security properties and identifies its strengths and weaknesses. One of the main features of the protocol is its ability to construct a session key even in the presence of malicious participants, which is achieved using a polynomial-based approach. However, constructing a polynomial with a high degree of complexity can be a burden on the scheme's efficiency. Additionally, the protocol is dependent on a public-key infrastructure, which may prevent it from being practically applicable in some settings.

2.5 Ring-Based Cooperation

Ring-based cooperation is a paradigm that organizes members of a group or organization into a circular or ring-like structure, with each member playing an equal role in essential management processes. In this approach, the group members employ a shared key or a set of keys to perform the encryption and decryption processes for the purpose of securing and accessing information within the group. Every participant takes on the roles of generating and distributing key management, thereby being accountable for safeguarding the shared key and handling their individual key. Whenever a node receives a message containing an intermediary key, it calculates the key and passes it on to another member.

Ring-based cooperation for group key management is a decentralized method that enables each member to share responsibility for the group's communication and data security. This method provides a high level of security because only authorized group members have access to the shared key. However, this strategy has limitations, such as traffic considerations and fault isolation. Unidirectional traffic can be disadvantageous, and if one participant does not cooperate, the entire process can be rendered ineffective.

The ING protocol [51, 52] requires a synchronous start-up and executes in (n - 1) rounds, where n is the total number of participants in the ring. In each round, every participant raises the previously received intermediate key value to the power of its own exponent (random secret) and forwards the result to the next participant in the ring. After (n - 1) successfully completed rounds, all n participants in the ring share a group key K.

The ING protocol, which can be seen as an extension of the two-party Diffie-Hellman protocol referenced as [53] , offers a decentralized solution for group key management, similar to other ring-based cooperation models. By ensuring that only authorized group members have access to the shared key, this approach guarantees a strong level of security. Nevertheless, it is important to acknowledge certain limitations of the ING protocol, such as the requirement for synchronous start-up and the potential vulnerability to attacks if any participant's security is compromised.

2.5.1 Ingemarsson et al., Protocol

Ingemarson et al.'s protocol, developed in 1982, is one of the earliest key management protocols designed to extend the public key distribution system proposed by Diffie-Hellman for conference group communication [51, 53]. The protocol addresses the challenges of secure communication and efficient key management in group settings. It enhances the sharing of cryptographic keys among participants, allowing for more than two participants (n>2) in a group. To achieve this, participants are required to establish a virtual ring topology or circular formation to facilitate key agreements. A signal is circulated along the virtual topology, passing from one station to another until it reaches the last station. This process is repeated for the generation and distribution of new keys. The transmission of messages in Ingemarson et al.'s protocol necessitates

simultaneous operation and execution, requiring (n - 1) rounds, where n represents the number of participants in the group. This protocol enables two or more members of a group to establish a shared group key, as indicated by the following formula.

$Round; k \in [1, n-1]$.

$M_i \rightarrow M_{(i+1)modn}: \left\{\alpha^{\Pi}(N_j | j \in [(i-k)modn, i])\right\}$

Example: $M = 4$. Transmitted messages from station 2:

Time Instant	Transmitted message
1	$\alpha^{R_2} modp$
2	$\alpha^{R_1+R_2}, \alpha^{R_1R_2} modp$
3	$\alpha^{R_0+R_1+R_2}, \alpha^{R_0R_1+R_0R_2+R_1R_2} modp$

Station 3 raises the first part of the last message to the power R_3 modulo p and obtains the conference key

$K^{(2)} = (\alpha^{R_0+R_1+R_2})^{R_3} \alpha^{R_0R_1+R_0R_2+R_1R_2} modp$

$\alpha^{R_0R_3+R_1R_3+R_2R_3+R_0R_1+R_0R_2+R_1R_2} modp$

All the participants agreed on a prime modulus and a generator in this case, P and $R_0, R_1 \ldots, R_{M1}$ in the range $\{1, p-1\}$. U_1 selects a private random number and calculates the number with mod p and sends the result publicly to U2 then U2 selects his private random number Where n is number of participants and integers $R_0, R_1 \ldots, R_{M-1}$, in the range $\{1, p-1\}$.Ri is the integer chosen randomly by station i and kept secret in that station.

Ingemarsson et al., protocol is considered inefficient for the following reasons:

- It starts with the initializing step to organize all members in a group.
- The conference key is established after computation in M_{i-1} rounds.
- The symmetrical nature of the protocol makes dynamic membership support a costly operation.

In the event of leaving any of the members or adding new members the entire process should be repeated for security purposes and to generate a new group key.

2.5.2 Group Diffie–Hellman (GDH) Key Exchange

The extension of Diffie-Hellman key exchange is a GDH key exchange protocol that supports group communications in which more than two members are participating. The basic strategy of GDH key management is the agreement of using a pair of two prime numbers, such as (q and g) ɜ $\mathbb{Z}_q^*$. The calculation process will be done separately by each member of the group in such a way that the first member calculates the first value and passes it on to the next entity in the group.

To generate a new set, each member uses its own secret number when it receives the set of intermediary values. A generated set will contain *n*-1 exponents and *n* intermediate values for a group of n members [37]. Initial key agreement of group key Diffie-Hellman is as follows if we have a group of three users:

User3 will receives the set from User 2 :

$\{g^{x2}, g^{x1}, g^{x1x2}\}$ and generates the set $\{g^{x2x3}, g^{x1x3}, g^{x1x2}, g^{x1x2x3}\}$

The key value for the upper example is g^{x1x2x3} . calculation of keys can be done by using key value using the formula (key = $g^{x1\cdots xn} \ mod\ q$).

Initiation: Let p be a prime and q a prime divisor of $p-1$. Let G be the unique cyclic subgroup of $\mathbb{Z}_q^*$ of order q, and let α be a generator of G.

Round $i(0 < i < n)$

1. M_i selects $r_i \in_R \mathbb{Z}_q^*$
2. $M_i \rightarrow M_{i+1}: \left\{ \alpha^{\frac{r_1 \dots r_i}{r_j}} \middle| j \in [1,i] \right\}, \alpha^{r_1 \dots r_i}$

Round n

1. M_n selects $r_n \in_R \mathbb{Z}_q^*$
2. $M_n \rightarrow ALL\ M_i: \left\{ \alpha^{\frac{r_1 \dots r_i}{r_j}} \middle| i \in [1,n] \right\}$

The resulting key is $\boldsymbol{\alpha^{r_1 \dots r_n}}$.

The protocol you provided is a key agreement protocol that enables a group of n parties (M_1, M_2, ..., M_n) to establish a shared secret key over an insecure communication channel.

The following is an explanation of the protocol :

1. Initialization:

- Choose a prime number p and a prime divisor q of $p - 1$.
- Find a cyclic subgroup G of $\mathbb{Z}_q^*$ of order q, and choose a generator α of G.

2. Round *i* (0 < *i* < n):

- Each party Mi selects a random number ri from $\mathbb{Z}_q^*$
- Mi computes a list of values of α raised to certain powers, as follows:
- For each j in {1, *i*}, compute α^((r_1 ... r_i)/ r_i).
- Mi sends the list of values {α^((r_1 ... r_i)/rj) | j in (1, i) and α^(r1...ri) to the next party, Mi+1.

3. Round n:

- The last party, Mn, selects a random number rn from $\mathbb{Z}_q^*$
- Mn computes a list of values of α raised to certain powers, as follows:
  - For each I in {1, n}, compute α^((r1...ri)/rn).
  - Mn sends the list of values {α^((r1...ri)/rn) | I in (1, n)} to all parties.

4. Key derivation:

- Each party Mi computes α^(r1...rn) using the values it received in the last round.
- The resulting key is α^(r1...rn).

The security of this protocol relies on the hardness of the discrete logarithm problem (DLP) in group G. In particular, an eavesdropper who intercepts the messages exchanged during the protocol would need to solve the DLP in G in order to recover the shared key, which is assumed to be computationally infeasible.

2.6 Hierarchy-Based Cooperation

The process of rekeying in this approach produces less overhead, a smaller number of broadcast messages, and a smaller number of rounds due to the strategy of distributing and sharing keys among subgroups. The key principle of this approach is to divide large-group communication into subgroups. In a dynamic environment, when there is a change in group size, the process of distributing new traffic encryption keys is not shared with all communication participants in

the topology. The protocols following Hierarchy-Based Cooperation are Octopus, Distributed Flat Table D-FT, Distributed One-Way Function Tree D-OFT, Skinny Tree Protocol STR, and Distributed Logical Key Hierarchy D-LKH.

2.6.1 Octopus Distributed Key Agreement Protocol

The Octopus protocol [54] also worked to extend the cryptography method proposed by Diffie-Hellman, it was presented in 1998 by researchers Becket and Wille. The key principle of this protocol is to divide large group communication into subgroups (if n represents the number of participants in a group communication, the number of subgroups produced by the Octopus protocol is n/4) as shown in figure 2.3. All the group members can calculate the group key by performing an II-party Diffie-Hellman exchange with their group node leader, $I_{subgroup} = \alpha^{u1u2...un/4}$, where u_i is the contribution from user I, and then the subgroups exchange their intermediary values [36, 37]. All four group node leaders then perform a IV- party Diffie-Hellman exchange the four subgroups leader or whatever in away the intermediary values of the first two subgroups will be exchange and create $\alpha^{I_a \cdot I_b}$ then the last two subgroups will be exchange and create $\alpha^{I_c \cdot I_d}$ later on the result of the first two subgroups will be exchange with the result of the second two subgroups using DH exchange method and the result will be $\alpha^{I_a \cdot I_b \cdot I_c \cdot I_d}$. After calculation by all the group node leaders, they send $_{\alpha}I_a \cdot I_b \cdot I_c \cdot I_d$ to their respective subgroups $_{\alpha}Ia \cdot Ib \cdot Ic \cdot Id / u_i$ where $i = 1 \cdots (n-4)/4$, and all members of the group are capable of calculating the group key.

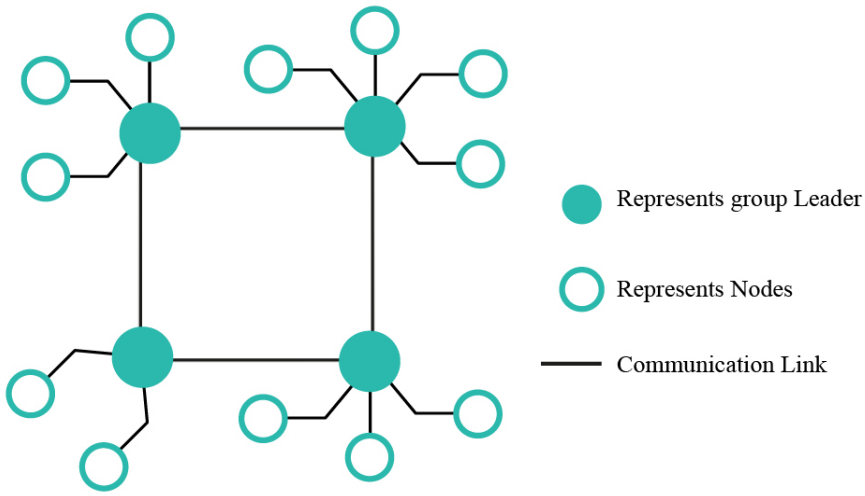

Fig. 2. 3: Visual Representation of The Octopus Protocol's Group Key Distribution Method.

The figure demonstrates the division of a large group's communication into smaller subgroups. The number of subgroups produced by the protocol is n/4, where n represents the number of participants in the group.

2.7 Encryption in Constrained Systems

Due to the nature and size of IoT devices, they usually have a limited number of resources. A consequence of this is that most IoT devices do not have the processing power or resources necessary for more robust encryption algorithms. Because encryption is still a necessary component of their functionality, lightweight encryption algorithms could be used. These algorithms could be implemented in software or through an integrated circuit (IC) in hardware. Each of these methods comes with an increase in cost for the IoT manufacturer because both methods require additional resources. Currently, there is no standard, and many IoT devices do not support encryption at all.

The National Institute of Standards and Technology (NIST) [55] has published a report on lightweight cryptography. In response to the growing need for cryptographic algorithms suitable for devices with limited resources, NIST has initiated the "lightweight cryptography initiative" aimed at developing such algorithms. These algorithms, referred to as 'lightweight cryptography,' are specifically designed to possess key characteristics such as low power consumption, compact code size, high operational speed, resilience against side-channel attacks, and secure key management. The primary purpose of these algorithms is to cater to applications on

the Internet of Things (IoT) and other domains where resource-constrained devices are prevalent.

2.8 Public Key Cryptography

The foundation of classic cryptography is the assumption that both the sender and the recipient have access to the identical secret key. The message is encrypted by the sender utilizing a secret key, and subsequently decrypted by the recipient using the same key. The term "symmetric cryptography" describes this kind of operation. The challenge is ensuring that the sender and the recipient use the same secret key. The difficulty in developing a secret-key cryptosystem has been the need to ensure the confidentiality of all keys.

The public-key cryptography introduced in 1976 by [17] and become a significant transformation with cryptographic systems. This technique involves the use of a pair of keys, namely the public key and the private key, which possess a mathematically related relationship. While the public key is publicly available, the private key remains confidential. Consequently, the need for the sender and receiver to exchange secret information is eliminated, as all communications rely exclusively on public keys. It is important to note that private keys are never transmitted or shared. Public-key cryptography enables the transmission of confidential messages using public information, such as the public key of the intended recipient, while decryption can only be achieved using the corresponding private key. Additionally, this system can be utilized to provide data or software authenticity and origin verification through the use of digital signatures.

In the realm of IoT devices, it is strongly advised to incorporate public-key cryptography to safeguard device security. This can be achieved by integrating the cryptographic algorithm directly into the device's firmware or integrated circuit (IC). However, this approach often entails additional expenses. Therefore, IoT manufacturers need to consider various factors, such as the robustness of the key and algorithm utilized, alongside the costs associated with implementing the required security measures. Employing a sufficiently strong key and algorithm can render it practically impossible to break through computational means, demanding an immense

amount of computing resources valued at trillions of dollars. This highlights the importance of effectively implementing security measures in IoT devices.

2.9 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) [56] is a contemporary class of public-key cryptosystems based on elliptic curves defined over finite fields and utilizing the Elliptic Curve Discrete Logarithm Problem (ECDLP). ECC provides encryption, digital signatures, and key exchange, which positions it as a prospective alternative to the conventional RSA cryptosystem. ECC offers a number of benefits over RSA [57], including reduced key sizes and signatures, rapid key generation, agreement, and signatures [58].

The utilization of ECC methods requires the selection of proper elliptic curves, which can provide varied levels of security, performance, and key length. The selection of an elliptic curve is critical since it has a direct impact on the overall security and efficiency of the ECC implementation [59]. Researchers have performed substantial research on elliptic curve cryptography, aiming to identify optimal curve parameters and methods that strike the correct balance between security and computing overhead, particularly in resource-constrained situations such as embedded computers.

Koblitz's pioneering work in 1987 laid the foundation for ECC and began the investigation of its cryptographic features. Since then, ECC has gotten a lot of attention from the research community because of its ability to address the limitations of standard public-key cryptosystems. Several studies have analyzed and evaluated the feasibility of ECC for embedded systems, taking into account issues such as power consumption, memory requirements, and computing efficiency. These investigations have underlined the benefits of ECC in fulfilling the demands of resource-constrained contexts while providing robust security. To implement discrete logarithm systems in ECC, cyclic subgroups of a point on an elliptic curve can be used. The elliptic curve E over a field of integers modulo p F_p can be denoted by an equation of the form:

$$y^2 = x^3 + ax + b \qquad (1)$$

Let *a, b* be arbitrary integers in F_p such that

$$4a^3 + 27b^2 \not\equiv 0(modp) \qquad (2)$$

A point (x, y) on the elliptic curve $\boldsymbol{E}$ such $x, y \in \mathbb{F}_p$ satisfies the equation if and only if it is a point on the curve. The set of all points on $\boldsymbol{E}(\mathbb{F}_{\boldsymbol{p}})$ is denoted by $\boldsymbol{E}(\mathbb{F}_{\boldsymbol{p}})$ Cyclic subgroups of a point $P(x, y)$ on $\boldsymbol{E}(\mathbb{F}_{\boldsymbol{p}})$ are useful for constructing cryptographic protocols based on the discrete logarithm problem.

ECC provides several groups of algorithms based on the math of elliptic curves over finite fields. These include digital signature algorithms, encryption algorithms, and key agreement algorithms. Two commonly used ECC digital signature algorithms are ECDSA (Elliptic Curve Digital Signature Algorithm) and EdDSA (Edwards-curve Digital Signature Algorithm). Popular ECC encryption algorithms include the ECIES (Elliptic Curve Integrated Encryption Scheme) and ECEG (EC-based ElGamal). Widely used ECC key agreement algorithms include ECDH (Elliptic Curve Diffie-Hellman), X25519, and FHMQV (Full-Hierarchy MQV). Several research studies have investigated the application of ECC in various fields, including wireless sensor networks and mobile devices. These studies have shown that ECC can provide a high level of security while utilizing limited resources, making it an efficient and effective cryptographic technique for resource-constrained devices. Therefore, ECC is considered a promising solution for secure communication in a wide range of applications.

2.9.1 Elliptic Curve Key Generation

The process of generating cryptographic keys using elliptic curves is referred to as elliptic curve key generation. The key generation involves using an elliptic curve $\boldsymbol{E}$ defined over a finite field of integers $\mathbb{F}_p$, with a point G in $\boldsymbol{E}(\mathbb{F}_{\boldsymbol{p}})$ having a prime order *n*. The cyclic subgroup generated by point G is represented as $G = \{\infty, G, 2G, 3G, \ldots\ldots (n-1)G\}$. The prime p, the equation of the elliptic curve $\boldsymbol{E}$, the point G, and its order n, are considered as public domain parameters. An elliptic curve key generation system uses four public domain parameters, which include a prime number *p*, an equation that defines the elliptic curve $\boldsymbol{E}$, a point *G* on the curve, and the order *n* of point *G*. A private key *d*, which is an integer randomly chosen from the interval {1,

$n-1\}$, is used to generate the corresponding public key Q on the curve $\boldsymbol{E}$. The public key is represented as a point:

$$Q = dG \qquad (3)$$

where the scalar multiplication of the private key d with the generator point G is performed according to equation (3). The elliptic curve discrete logarithm problem refers to the difficulty of determining the private key d given the public domain parameters and the public key Q.

2.10 Routing Protocol

Data transmission flow is one of the most important aspects of WSN and is given a lot of attention. A collection of protocols collectively referred to as routing regulates the direction in which data is sent from one point to another across a network. Many different factors, such as energy consumption, coverage area, and others, should be taken into account when developing routing protocols for WSNs. On the basis of network structure, WSN routing protocols can be divided into three groups: flat, location-aware, and hierarchical [60].

Flat routing directs all traffic to the base station, resulting in significant energy consumption. In this setup, each network node has the same function and can communicate with the sink node or base station through intermediate nodes. However, due to its limitations, this routing strategy is not recommended in large-scale systems. Hierarchical routing systems, in contrast to flat routing, partition the network into clusters, where each cluster has a cluster head responsible for communicating with the base station. This saves energy by reducing the number of nodes that connect directly with the base station, resulting in a longer network lifetime and improved scalability. However, hierarchical routing protocols require more complicated management techniques and are susceptible to cluster head failure [61, 62].

Another main category of routing protocols is location-based routing, in which each node in the network has a unique location and uses this information to determine routing decisions. This approach can enhance network efficiency by lowering communication overhead and minimizing the routing path, but it requires accurate location information and can be affected by location errors and mobility [63]. The use of machine learning techniques, such as

reinforcement learning and neural networks, to optimize routing decisions based on changing network conditions and application requirements has recently emerged as a new trend in wireless sensor network (WSN) routing. These approaches can provide better adaptability and fault tolerance, but it require more computational resources and may pose challenges in terms of security and privacy [64]. Figure 2.4 depicts the different subcategories of wireless sensor network (WSN) routing protocols.

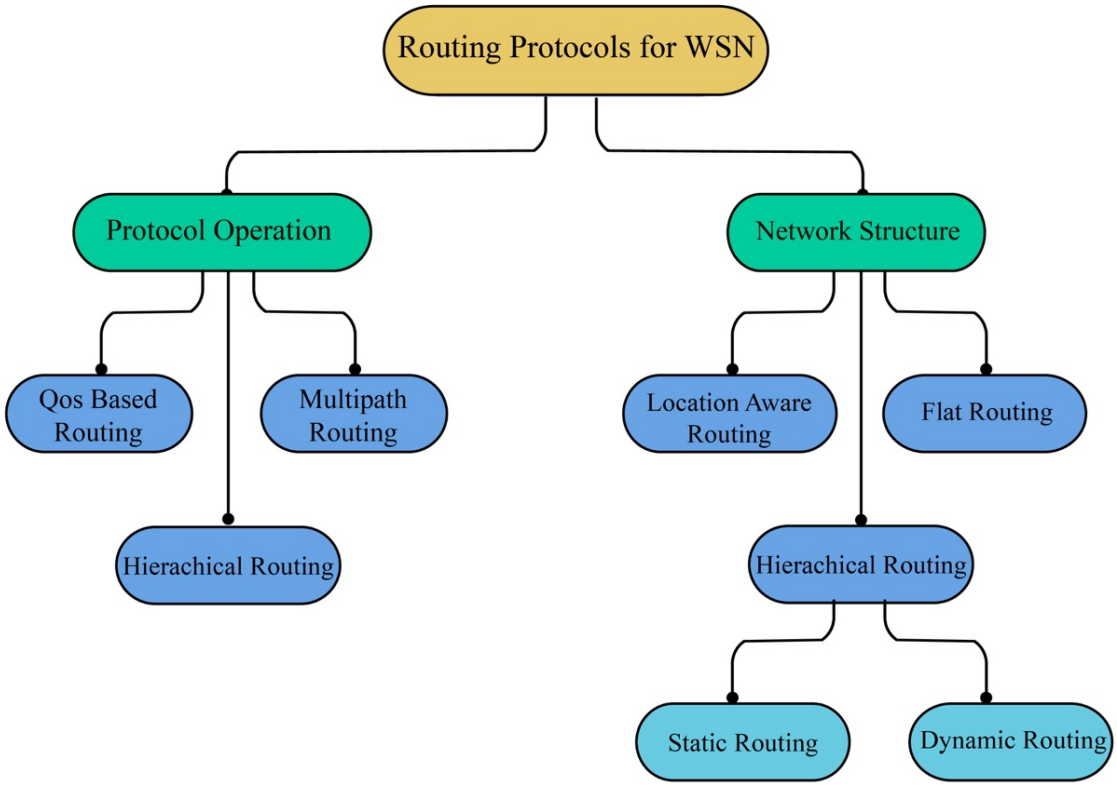

Fig. 2. 4: Routing Protocol Classification [62].

This diagram depicts the classification of wireless sensor network routing protocols. The diagram is split into two parts: protocol operation and network structure. The protocol operation section describes the various types of routing protocols based on their operation, such as QoS-based routing, query-based routing, and multipath routing. The network structure section describes the various types of routing protocols based on their network structure, including flat routing, location-aware routing, and hierarchical routing. Static routing and dynamic routing are further classifications of hierarchical routing. This classification helps to provide a better understanding of the various routing protocols available for WSNs and their specific features and capabilities.

Table 2. 1: Comparison of Routing Techniques

Routing Technique	Energy Consumption	Adaptability	Fault Tolerance
Flat Routing	Limited	Limited	Limited
Hierarchical Routing	High	Limited	Limited
Location-based Routing	Limited	High	Limited
Machine Learning-Based Routing	Low	High	High

The "Comparison of Routing Techniques" table compares several routing techniques based on their energy consumption, adaptability, and fault tolerance. The table categorizes routing strategies into four categories: flat routing, hierarchical routing, location-based routing, and machine learning-based routing.

Flat routing is distinguished by low energy usage, adaptability, and fault tolerance. On the other hand, Hierarchical routing consumes more energy but is less adaptable and fault-tolerant. Location-based routing consumes less energy, is more adaptable, and has a lower fault tolerance.

Machine learning-based routing, on the other hand, has low energy consumption, high adaptability, and high fault tolerance, considered an optimum solution for some types of applications. However, it is also important to consider other factors such as computational resources, security, and privacy when deciding on a routing technique [65].

Several routing methods have been developed to optimize energy consumption and energy balance in sensor networks, with the goal of extending the network lifetime and enhancing overall system performance.

The Low Energy Adaptive Clustering Hierarchy (LEACH) is a well-known routing protocol proposed in the paper [29]. This protocol uses a hierarchical structure routing strategy to improve the robustness and scalability of networks that fluctuate in size dynamically. The protocol suggests using a random number between 0 and 1 to assist in the selection of the cluster head (CH). The threshold value T (n) is calculated using the following formula:

$$T(n) = \begin{cases} \dfrac{p}{1 - p * \left(r \bmod \left(\frac{1}{p} \right) \right)}, if node has not been a cluster head \in the past \dfrac{1}{p} rounds \\ 0, if node has already been a cluster head \in the past \dfrac{1}{p} rounds \end{cases}$$

The protocol selects cluster heads from the nodes randomly, using a threshold value determined by a formula that incorporates the desired percentage of nodes to become cluster heads (denoted by parameter p), the current round number (denoted by r), and the set of nodes that were not chosen as cluster heads in the last 1/p rounds (represented by G). The parameter p is a value between 0 and 1, and the selection of cluster-heads is performed randomly to ensure a fair distribution of energy consumption among the nodes. The protocol operates in rounds, with each round consisting of a setup phase and a steady-state phase. In the setup phase, cluster heads are selected, and clusters are formed. In the steady-state phase, each node sends data to its corresponding cluster head, and the cluster head aggregates the data and transmits it to the base station. The parameter 1/p is used to determine the number of rounds between successive selections of cluster heads, allowing for dynamic clustering and energy consumption distribution.

The primary purpose of the LEACH protocol is to maintain a balanced distribution of energy consumption across all nodes in the network. The protocol achieves this through a hierarchical structure that consists of three members: the sink, cluster head, and cluster nodes.

The sink acts as the central station, collecting data from all nodes in the network and uses this data for further processing. To prevent the same node from becoming the cluster head several times, one node is randomly chosen from the other nodes in the network to act as the cluster head.

The cluster head is responsible for designating nodes within its cluster by sending a request message to all geographically nearby nodes. Any nodes that send an acknowledgement message back to the cluster head are accepted as members of the cluster. Each cluster member then sends its data to the cluster head, which aggregates the information and transmits it to the sink.

Wu et al. [52] combined two strategies to provide an energy-aware routing algorithm for solar wireless sensor networks: Grouping of Unequal Clusters Using Energy Harvesting and Optimal Adaptive Performance Routing (EHGUC-OAPR). EHGUC first selects multiple groups of nodes to operate as cluster heads based on a weighted sum that takes energy harvesting rate, distance, and other variables into consideration. This technique decreases the cluster size and separation from the base station, reducing the overall energy consumption of the clusters, allowing for more energy to be stored and used for inter-cluster data analysis. The optimal adaptive performance routing algorithm (OAPR) then determines the next hop based on each node's ability to sustain energy to ensure accurate packet transmission. Table 2.2 summarizes the key features, advantages, and disadvantages of several cluster-based routing protocols for wireless sensor networks. The table provides a comparison of different protocols, helping researchers and practitioners choose the appropriate protocol for their needs [66].

Table 2. 2: Comparison of Cluster-Based Routing Protocols for Wireless Sensor Networks

Protocol	Key characteristics	Advantages	Limitations	Reference
Low-energy adaptive clustering hierarchy (LEACH)	Uses clustering to divide a network into smaller groups with cluster heads for communication with the BS. Implements local compression to reduce global communication costs in terms of energy consumption and transmitted messages.	Provides localized coordination and control for cluster setup and operation, randomized rotation to elect cluster heads among nodes, energy efficiency, simplicity, and load balancing, which can expand the network lifespan.	Not applicable to time-constrained applications or large-scale networks due to single-hop communication.	[29]
Hybrid and energy efficient distributed (HEED)	Forms clusters of equal size and selects cluster heads based on primary and secondary network parameters.	Minimizes intra-cluster communication energy consumption, achieves well distributed cluster nodes in the network.	Increases control message overhead during the cluster formation phase.	[67]
Cluster-based even driven routing protocol (CERP)	Forms clusters based on various events and calculated the shortest path using distance-based link cost.	Limits energy consumption by providing the shortest distance between cluster heads and the base station.	May isolate nodes in the network and may result in unequal cluster formation due to event occurrence.	[68]
Dijkstra-based weighted sum minimization (DWSM)	Uses a multi-objective weighted function to calculate link cost between nodes and investigates the impact of varying weighting factors on wireless mesh network performance.	Flexible in selecting different metrics for optimization, such as delay or capacity, based on the requirements of the specific application. The method may be suitable for implementation on constrained devices with limited processing power and memory.	The proposed method is focused on static WMNs and may not be applicable to dynamic or mobile networks. In addition, the performance of the proposed method is dependent on the weighting factor, which	[69]

			needs to be carefully chosen for optimal results.	
Hierarchical unequal clustering fuzzy algorithm (HUCFA)	Divides the network area into three horizontal layers and uses fuzzy logic for cluster head selection.	reduced energy consumption, improved energy efficiency, and extended network lifetime. The fuzzy-logic-based CH selection scheme provides flexible and adaptive selection of CHs, leading to balanced energy consumption across the network. The hierarchical clustering approach reduces communication overhead and network congestion, improving communication efficiency.	Provides inefficient results when node mobility increases due to linguistic variables.	[66]
Fuzzy maximum lifetime (FML)	Uses a fuzzy membership function to calculate link weight and selects the minimum weighted path via the Dijkstra algorithm.	Maximizes network lifetime by taking the residual energy of the source node into account and considers node mobility.	Inapplicable to time-constrained applications.	[70]
Unequal Clustering approach using Fuzzy logic (UCF)	a. The UCF algorithm selects nodes with the highest remaining energy in each region as candidate cluster heads (CHs) and employs fuzzy logic to adjust the cluster radius of CH nodes based on local information. b. UCF uses a distributed clustering algorithm that primarily selects CHs based on the	a. UCF balances energy consumption across the network, prolonging the network's lifetime and improving energy conservation. b. The algorithm achieves load balancing by constructing unequal clusters, reducing the appearance of hot spots, and diminishing sensing coverage. c. UCF exhibits low clustering energy overhead due to a decrease in the number	a. The proposed algorithm assumes stationary sensor nodes are randomly distributed in the field and does not consider mobile sensor nodes. b. While the simulation results demonstrate the effectiveness of UCF, further real-world implementation and validation are required to assess its	[71]

	residual energy of nodes, avoiding the generation of orphan nodes that do not belong to any cluster. c. UCF considers the local density of nodes to determine the cluster radius, addressing the unbalanced energy consumption issue caused by random node distribution.	of messages exchanged for constructing the clusters. d. Simulation results show that UCF outperforms well-known clustering algorithms such as M-LEACH, HEED, and DUCF in terms of network lifetime, load balancing, and energy efficiency.	performance in practical scenarios.	
Low-energy adaptive clustering hierarchy-dynamic threshold (LEACH-DT)	Uses dynamic energy threshold values to select cluster heads, resolves the reallocation time slot problem among candidate and current cluster head nodes, balances the energy consumption of nodes in the network, and	Uses dynamic threshold values to select cluster heads, which balances the energy consumption of nodes in the network and resolves the reallocation time slot problem among candidate and current cluster head nodes. Addresses the uneven distribution of cluster heads, which may lead to a hot-spot problem in the network.	Uneven distribution of cluster heads, which may lead to a hot-spot problem in the network. Does not consider the residual energy of nodes in the network, which may lead to premature node failure and reduce the network lifetime. The selection of cluster heads based on the dynamic threshold may not guarantee an optimal solution for energy consumption and may result in suboptimal performance in some cases.	[72]

The table presents an overview of several clustering routing protocols used in wireless sensor networks. Each protocol has its own key characteristics, advantages, and limitations. LEACH uses clustering to divide the network into smaller groups and implements local compression to reduce energy consumption. HEED forms clusters of equal size and selects cluster heads based on primary and secondary network parameters. CERP forms clusters based on various events and calculates the shortest path using distance-based link costs. DWSM uses a multi-objective weighted function to calculate the link cost between nodes. HUCFA divides the network area into three horizontal layers and uses fuzzy logic for cluster head selection. SPFL uses a pool manager node to select the shortest path and proposes a fuzzy logic function for data routing. FML uses a fuzzy membership function to calculate link weight and selects the minimum weighted path via the Dijkstra algorithm. DUCF implements an unequal clustering mechanism and determines cluster size and cluster heads using fuzzy logic. LEACH-DT uses dynamic energy threshold values to select cluster heads and balances the energy consumption of nodes in the network.

The presented clustering routing protocols aim to increase network lifespan, minimize energy consumption, and balance energy consumption among nodes. Each protocol has its strengths and weaknesses, and the selection of the protocol depends on the application requirements and network characteristics.

2.11 Constrained Devices

According to RFC 7228 from the Internet Engineering Task Force (IETF), the Internet of Things (IoT) and Wireless Sensor Networks (WSN) primarily consist of constrained devices. A constrained device lacks one or more features that are standard for most Internet nodes due to resource limitations, such as cost or physical constraints like size, weight, or available power. Due to these restrictions, energy and network bandwidth optimization are crucial factors in all design specifications.

The communication capabilities of constrained devices are also limited. Even when communication is available, encryption is often not implemented due to the limited processing power of these devices, especially Class 0 devices. This lack of encryption is one of the vulnerabilities listed by the Open Web Application Security Project (OWASP).

Sensor: A sensor is a device that measures a physical quantity and converts that measurement reading into a digital representation. Typically, another device receives this digital representation and transforms it into useful data that humans or intelligent machines can use. The use of sensors in the IoT has enabled a new paradigm of business intelligence by allowing connected physical objects to communicate with each other and external systems, interpret their environment, and make intelligent decisions. The wide variety of sensors available can be grouped into different categories, including active or passive, invasive or non-invasive, contact or no-contact, absolute or relative, area of application, how sensors measure, and what sensors measure. The most practical classification for sensor applications in an IoT network is based on what physical phenomenon the sensor is measuring. Sensors are utilized in many fields, including precision agriculture, which uses technical advances such as GPS, aerial imagery, robots, real-time analytics, and artificial intelligence to enhance farming practices in terms of efficiency, sustainability, and profitability [73].

Smart Sensors: The term 'smart sensor' was first used in the mid-1980s, and since then, a wide variety of gadgets have adopted the name. These gadgets are made possible by several different types of semiconductors, including microcontroller units (MCUs), digital signal processors (DSPs), and application-specific integrated circuits (ASICs), which provide the necessary intelligence [74].

Smart sensors are a critical component of IoT devices, although sensors have existed long before the IoT. Initially, they only provided a visual indication of the measured data to the user. With customized protocols and interfaces, certain sensors were able to communicate with other devices. However, even today, many sensors require a gateway to collect data before it can be used in an IoT setting. With an embedded microprocessor, smart sensors can connect directly with a monitoring system. They can also self-diagnose if a problem arises, making them particularly useful for remote monitoring and control applications.

Actuators: Actuators are devices that receive a control signal and produce a physical effect, such as motion or force. They complement sensors, which sense and measure variables in the physical world and convert them into electric signals or digital representations. The interaction between sensors, actuators, and processors is similar to the way the human senses and nervous system work together, as shown in Figure 2.5. Sensors detect changes in the physical environment and collect data, which is then transmitted to a processing unit

for analysis. The processing unit uses algorithms to interpret the data and make decisions, which are then communicated to actuators.

Actuators are categorised according to their type of motion, power, binary or continuous output, application area, and energy type. The most common classification is based on the type of energy, which includes mechanical, electrical, electromagnetic, hydraulic, pneumatic, smart material, and micro- and nano-actuators. Sensors and actuators can be utilized to solve common problems by transforming sensor data into actionable insights that actuators can act on. For example, in precision agriculture, smart sensors that evaluate soil quality can be connected with valve actuators to deliver a highly optimized and custom environment-specific solution.

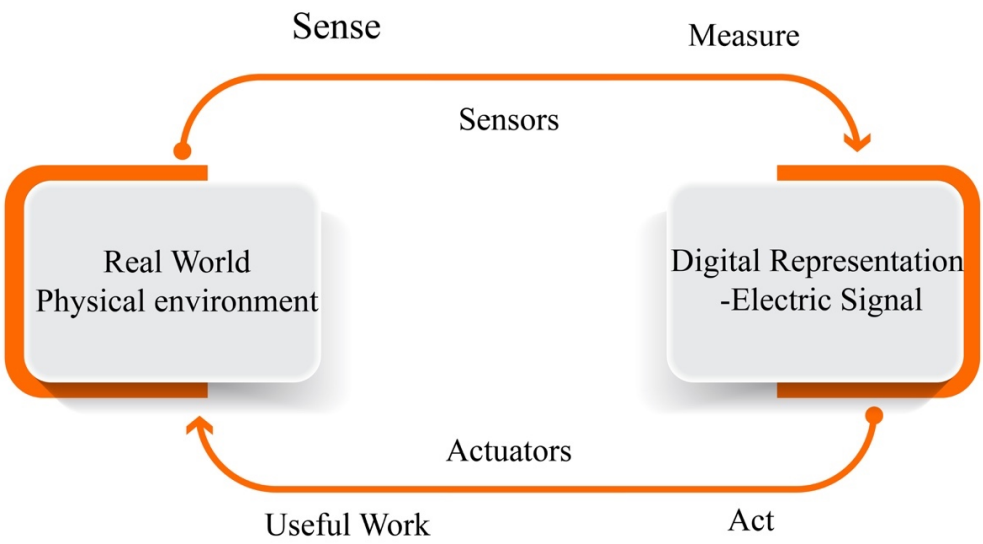

Fig. 2. 5: Interaction of Actuators And Sensors With The Physical Environments [73].

Embedded Devices: An embedded device is a product that has a computing system specifically designed to perform a particular function. An embedded device's operating system is optimized to execute a single application, ensuring efficiency and reliability [75].

Low-cost embedded devices are used in a variety of applications, such as automated teller machines (ATMs), point-of-sale (POS) terminals, and smart appliances such as dishwashers and refrigerators. These devices can be either smart, capable of connecting to the

internet and communicating with other devices, or dumb, having limited functionality and no internet connectivity [76].

There are numerous applications for low-cost embedded devices, including automated teller machines (ATMs), point-of-sale (POS) terminals, and smart appliances such as dishwashers and refrigerators. They are able to communicate with other devices through the internet, enabling a vast array of possibilities. However, dumb embedded devices are not connected to the internet and have limited functionality, such as sensors and actuators that perform a single task [77].

Prototyping: Raspberry Pi and Arduino are popular prototype platforms for embedded systems, particularly in robotics, sensing, and weather monitoring. The Raspberry Pi has impressive computing power and a wide variety of ports, but it needs a full operating system to function and lacks standard input devices like keyboards and mice. The Arduino, on the other hand, is a single-board microcontroller that can be readily programmed to carry out a variety of tasks. The program is then compiled and sent to the Arduino's non-volatile flash memory.

Despite their strengths, both the Raspberry Pi and Arduino have their own weaknesses. For instance, the Raspberry Pi has a limited lifespan and requires a reliable power supply connector, while the Arduino does not have an onboard ADC or EEPROM, FRAM or SPI Flash for data logging applications. However, both devices have a large online community with readily available examples and community support [78]. Table 2.3 is a reference table that provides information on the RAM data size and flash storage code size for three different classes of constrained devices. These classes include Class0/C0, Class1/C1, and Class2/C2. The table also includes a brief description of each class and their primary use cases [79].

TABLE 2. 3: Constrained Device Classes as Defined by RFC 7228 [79].

Name	Data Size (RAM)	Code Size(Flash Storage)	Description
Class0,C0	<10 KB	<100 KB	A gateway used for basic communication requirements.
Class1,C1	~10 KB	~100 KB	Use the protocol stack for IoT devices that support CoAP. Interact with other devices without using a gateway.
Class2, C2	~50KB	~250 KB	These devices handle both the standard IPV4 and IPV6 protocols. They work in the same way as other network devices.

The table categorizes nodes into three types depending on their processing and storage capacity, power supply, and ability to implement an IP stack and associated security mechanisms. Class 0 nodes have severely limited resources, with less than 10 KB of memory and less than 100 KB of flash processing and storage capability. They are often powered by batteries and cannot implement an IP stack or security mechanisms directly. Push buttons that transmit 1 byte of information upon changing status are examples of Class 0 nodes that are suited for LPWA wireless technology.

Class 1 nodes have significantly greater storage space and memory than Class 0 nodes, with approximately 10 KB of RAM and 100 KB of flash. Using a full IP stack is difficult for communication with nodes, but optimized stacks like CoAP make it possible for them to communicate with nodes. Without a gateway, they able to interact with the network and perform security functions. Class 1 nodes include devices such as environmental sensors.

Class 2 nodes operate entire IP stacks on embedded devices with over 50 KB of memory and 250 KB of flash. They can be fully integrated into IP networks, and examples include smart power meters [73].

2.12 WSN, IoT, WoT, M2M, and CPS

A wireless sensor network (WSN) is a collection of sensors distributed across a geographic area that can sense or regulate physical characteristics such as temperature, humidity, sound, light, and others. These sensors communicate with each other and with the base station or sink

through wireless channels. Sensor nodes have limited resources, including energy, memory, and CPU capacity, and typically include a power unit, a processing unit, one or more sensing units, a transceiver, an antenna, and optional components such as a position-finding system, a power generator, and an actuator. The volume of sensor nodes can vary greatly, ranging from cubic nanometres to cubic decimetres.

The location of sensor nodes within a network can either be known or unknown, and the network's topology is established through actual or logical communication between network nodes and other devices. Different topologies might exist depending on the network and node tasks. The efficiency and reliability of data transmission are crucial in wireless sensor networks (WSNs), and routing protocols play a vital role in deciding how data is transmitted over the network. WSN routing methods must take into account aspects like energy consumption, coverage area, and other relevant concerns. Routing protocols are categorized into three types based on network topology: flat, location-aware, and hierarchical [60].

Over time, several subfields of research have emerged to investigate wireless sensor networks (WSNs), driven by the growing demand for WSN applications. These subfields encompass Ad Hoc wireless networks, the Internet of Things (IoT), machine-to-machine (M2M) communication, Cyber Physical Systems (CPS), and the Web of Things (WoT). Each of these fields is interconnected and has evolved in response to the distinct challenges and prospects offered by WSNs. In the following sections, we will provide a succinct overview of each field and its correlation with WSNs.

As the demand for WSN applications has grown, various subfields of research have emerged to investigate wireless sensor networks (WSNs). Ad Hoc wireless networks, the Internet of Things (IoT), machine-to-machine (M2M) communication, Cyber Physical Systems (CPS), and the Web of Things (WoT) are examples of these subfields. Each of these sectors is interconnected and develops in response to the distinct challenges and opportunities presented by WSNs.

Sensors may be passive or active devices employing a variety of sensing principles, including mechanical, chemical, chromatographic, magnetic, biological, fluidic, thermal, electrical, optical, ultrasonic, and mass sensing. For WSNs to be implemented in a practical and cost-effective manner, sensors must be small, low-cost, robust, dependable, and sensitive. Sensor node hardware designs can range from nodes connected to a LAN and connected to permanent power

sources to nodes communicating via a wireless multi-hop RF radio powered by tiny batteries. Very-large-scale integration (VLSI), integrated optoelectronics, and nanotechnology are three areas of study that are influencing the development of sensor nodes, with ongoing research focused on building advanced WSNs and motes at the cubic millimeter scale [75]. Figure 2.6 depicts the typical architecture of a wireless sensor network (WSN), which consists of multiple sensor nodes, a sink, a gateway, and the Internet.

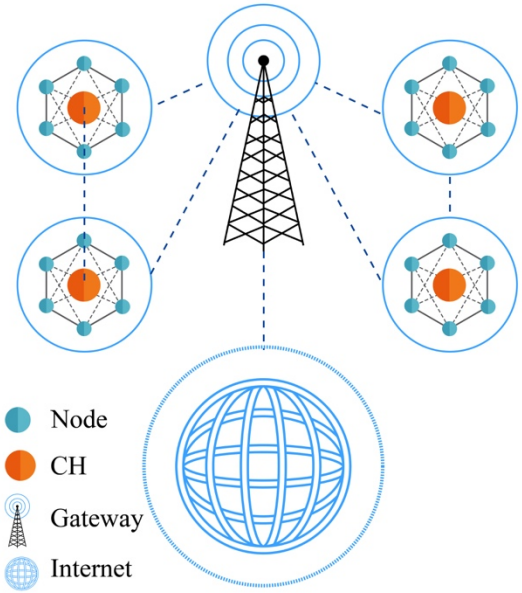

Fig. 2. 6: Conventional Architecture of a Wireless Sensor Network (WSN).
The diagram depicts a typical WSN design, which includes numerous sensor nodes, a sink, a gateway, and the internet. The network is divided into clusters, with each cluster head sending data from its cluster to the base station. The data is subsequently transmitted to the internet via the base station for further processing.

2.12.1 Internet of Things (IoT)

The Internet of Things (IoT) is a promising technology that has emerged as an offshoot of wireless sensor networks. Scholars have defined IoT in various ways; Vermesan et al. [80] defined IoT as an interaction between the physical and digital worlds, where the digital world interacts with the physical world using a plethora of sensors and actuators. Peña-López et al. [81] provide an alternative characterization of the Internet of Things (IoT) as a paradigm wherein the integration of computing and networking capabilities is incorporated into a wide range of conceivable objects. According to [82], the IoT is described as a technological

revolution that enables the connectivity of numerous components, including sensors and smart devices, to the Internet, forming a vast network.

The Internet of Things (IoT) can refer to a large and complex system comprised of numerous sensors, actuators, and gateways. The connection between IoT devices and gateways often relies on a wide range of protocols, enabling seamless communication. Gateways, in turn, establish connections to the internet and cloud applications using an equally diverse set of protocols. To assure the security of an Internet of Things (IoT) system, it is essential to identify potential system vulnerabilities. Segmenting the system into different functional areas can simplify its complexity and provide a useful foundation for understanding how it works.

The European Telecommunications Standards Institute (ETSI) [83] developed an architecture for machine-to-machine (M2M) communications, that includes IoT devices. The ETSI model divided into three domains: machine-to-machine (M2M), the network (communication), and application. The primary objective of this model is to provide a standardized framework for understanding the placement of various protocols and standards in an IoT system. The three domains in the ETSI model are depicted in Figure 2.7 in the section. The ETSI model supports interoperability and integration between devices and applications in the IoT ecosystem by providing a common framework for understanding the placement of protocols and standards in IoT systems. This standardization enables efficient communication and enhances the overall performance and functionality of IoT networks.

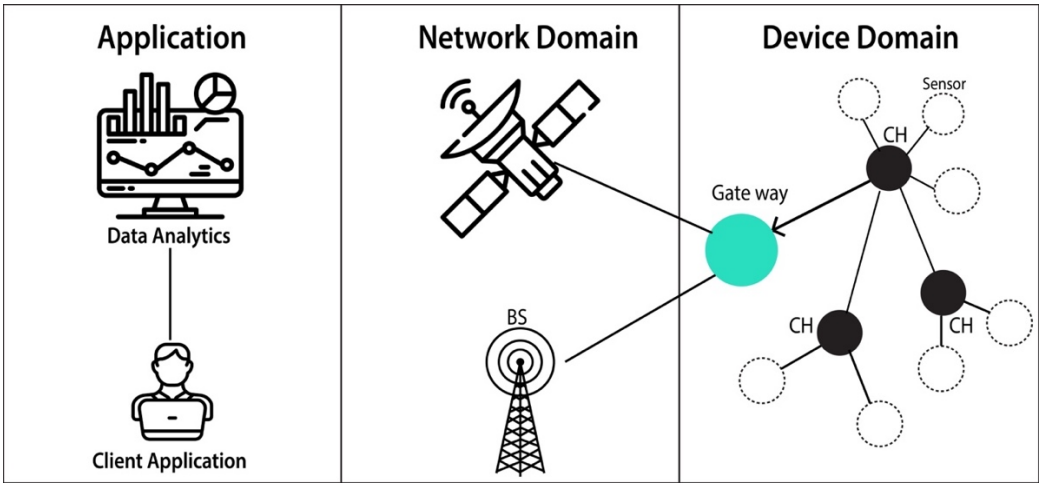

Fig. 2. 7: ETSI Reference Architecture [83].

The depicted diagram, labeled ETSI Reference Architecture, provides a comprehensive overview of the various standards and protocols utilized within an IoT system. These protocols and standards operate across many layers of the IoT architecture, facilitating connectivity and enables seamless communication between devices. These standards and protocols can be classified into three primary domains based on the IoT architecture layers in which they are implemented: application, communication, and device layers [73].

Figure 2.8 provides a comprehensive overview of the diverse standards and protocols utilized within an IoT system. It categorizes these standards and protocols based on the layers of the IoT architecture where they are implemented. The protocols and standards showcased in the figure are divided into three distinct layers. The application layer encompasses protocols primarily concerned with higher-level functionalities; the communication layer involves protocols responsible for data transmission and network communication; and the device layer consists of protocols relevant to the devices and sensors utilized in the IoT system. The following sections provide an overview of the protocols found in each layer [84].

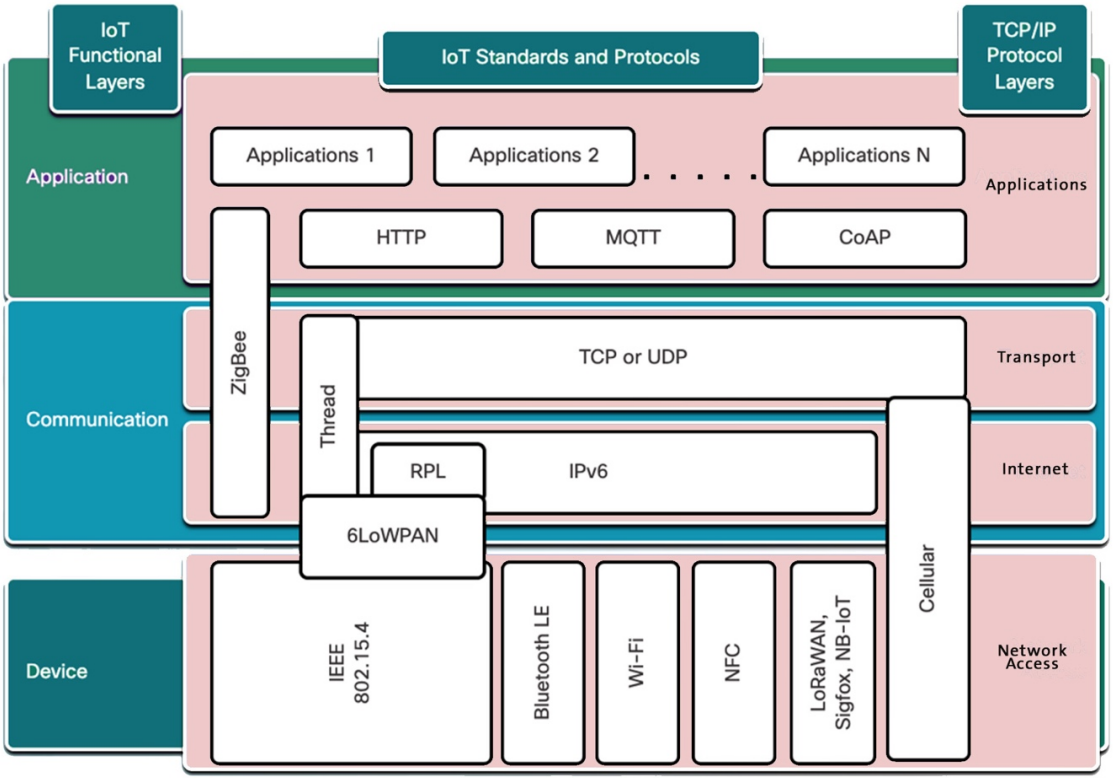

Fig. 2. 8: IoT Protocol Model [84].

This diagram depicts a comprehensive view of the diverse standards and protocols employed in an IoT system, organized by the layers of the IoT architecture in which they are implemented. There are three layers of protocols and standards: application, communication, and device.

Application Domain :

- Zigbee: A suite of protocols that use low-power digital radios based on the IEEE 802.15.4 wireless standard, primarily used in the application and communication layers of the IoT architecture.
- HTTP/HTTPS: robust application protocols used for getting and posting data.
- MQTT: The Message Queue Telemetry Transport (MQTT) protocol is employed in the middle to upper layers of the IoT, utilizing a broker-based architecture. MQTT enables the sensor to operate as a publisher of information, while the application requiring the data can act as a subscriber. Any intermediary system can serve as a broker to relay information between the publisher and subscriber(s). MQTT operates over TCP, leading to an MQTT client maintaining a continuous connection with the broker. However, this feature may prove to be a constraint in settings with high loss rates or inadequate computing resources.
- A lightweight publish and subscribe messaging protocol designed for resource-constrained devices that use TCP.
- CoAP: The Constrained Restful Environments (CoRE) working group of the Internet Engineering Task Force (IETF) developed the Constrained Application Protocol (CoAP) as a response to the limitations of web-based protocols. CoAP utilizes some methods that resemble those of HTTP, including Get, Post, Put, and Delete, albeit with a reduced list that curtails the size of the header. Unlike HTTP, which typically employs TCP, CoAP operates on UDP. Additionally, CoAP introduces a valuable feature absent in HTTP observation. Through this functionality, CoAP allows for the real-time streaming of state changes without necessitating the receiver to initiate requests for such modifications.

Communication Domain:

- Thread: A standard for home automation that uses IPv6 for routing on top of an IEEE 802.15.4 wireless network.
- TCP: is a reliable transport protocol that guarantees data delivery through synchronization and acknowledgment messages.
- UDP: is a lightweight, unreliable transport protocol that has no mechanism for guaranteed data delivery.

- RPL: A Routing Protocol for Low-Power and Lossy Networks that uses IPv6 to address devices in lossy networks.

Device Standards:

- IEEE 802.15.4: A standard for low-rate wireless personal area networks (LR-WPANs) meant for low-cost, low-speed devices.
- BLE: A wireless personal area network protocol that reduces power consumption without sacrificing range.
- Wi-Fi: A collection of IEEE 802.11 standards for wireless local area networks operating at 2.4 GHz and 5 GHz frequencies.
- NFC: A collection of protocols for device-to-device communication when devices are in close proximity.
- Cellular: All cellular technologies covered by the 3GPP, such as 4G, LTE, and 5G.
- LPWAN protocols: such as LoRaWAN, Sigfox, and NB-IoT are designed to transmit small data payloads over long distances at low transfer rates.
- 6LoWPAN: is an IETF standard for IPv6 Low-power Wireless devices in a Personal Area Network that provides a way for IPv6 to conform to the IEEE 802.15.4 standard.

The Internet of Things (IoT) encompasses various topologies that can be classified into three primary categories: star, mesh, and peer-to-peer. The star topology is frequently employed in both short-range and long-range technologies, including cellular, LPWA, and Bluetooth networks. This particular topology employs a single central base station or controller for the purpose of facilitating communication with endpoints. In the context of medium-range technologies, a hybrid approach incorporating star, peer-to-peer, or mesh topologies is commonly employed. Peer-to-peer topologies facilitate communication between devices within proximity, enabling any device to establish a connection with any other device. On the other hand, mesh topologies enhance long-range communication by employing intermediary nodes to relay traffic on behalf of other nodes, thereby mitigating the need for high transmit power.

2.12.2 Web of Things (WoT)

The concept of the Web of Things (WoT) involves the application of the fundamental principles of the World Wide Web to the realm of the Internet of Things (IoT). This extension aims to facilitate the establishment of consistent and effortless interactions between various devices and services. The framework is constructed based on the fundamental principles of interoperability, security, and accessibility. This design allows developers to create novel applications and services that effectively utilize the extensive volumes of data produced by Internet of Things (IoT) devices. The World of Things (WoT) encompasses a standardized collection of principles and protocols that facilitate effective communication among various interconnected devices, as well as tools and frameworks for building and deploying IoT applications [85, 86].

The WoT is founded upon the REST (Representational State Transfer) architectural style, which facilitates the extension of web tools and techniques to physical entities by incorporating web servers into smart objects. This facilitates the establishment of an inclusive ecosystem of digitally augmented objects that can be used to create applications using standard web languages and tools. However, the preservation of confidentiality and protection of personal data produced by IoT devices remains a significant challenge that requires the development of standards that can be adopted for various devices and platforms.

It is required to address challenges such as privacy, security, and standardization in order to unlock the full potential of the WoT in developing smart homes, cities, and businesses that optimize various processes through the massive volumes of data created by IoT devices. The WoT depends on OSI model Layer 7 protocols and techniques, including as REST, HTTP, JSON, microdata, and Web Sockets, to enable smooth communication between devices, making it a powerful platform for developing creative IoT applications [86].

2.12.3 M2M

Machine-to-machine (M2M) communication refers to a communication technology where a large number of intelligent devices can autonomously communicate with each other without direct human intervention. This technology enables collaborative decision-making among machines and helps achieve better cost efficiency and time management. M2M communication is

rooted in supervisory control and data acquisition (SCADA) systems, where sensors and devices are connected through wired or radio frequency networks to monitor and control industrial processes [87].

The Cisco Annual Internet Report predicts that M2M connections will account for 50% of all globally connected devices and connections by 2023, up from 33% in 2018. The rising popularity of M2M applications in a variety of industries, such as connected homes, connected automobiles, and connected cities, is propelling the rise of M2M communications [88].

The growth of M2M communications is strongly influenced by the low cost and pervasive connectivity of IP connected devices, such as sensors, monitors, and actuators. These devices play a vital part in facilitating the expansion of interconnected and interoperable services, which are collectively referred to as the Internet of Things (IoT). M2M technologies have a wide array of applications and are characterized by their diversity in device functionalities and other requirements. Consequently, developing a flexible M2M architecture capable of accommodating current and future technologies while ensuring interoperability, confidentiality, privacy, and reliability poses a significant challenge that requires collaborative efforts and coordination among cross industry groups at an international level.

M2M communication introduces potential challenges when it comes to machines communicating with each other. This is mainly due to the fact that billions of devices are constantly communicating for different purposes. As a result. This leads to congestion and overload in networks, which in turn generates various types of data traffic. The critical challenges that arise in M2M communications include energy efficiency, reliability, security, ultra-scalable connectivity, heterogeneity, and quality of service (QoS). In order to tackle these challenges standard developing organizations (SDOs) like 3GPP, ETSI, oneM2M and the IETF have taken steps to promote standardization activities focused on addressing issues related to M2M communications [89].

2.12.4 Cyber-Physical Systems

The CPS framework represents a novel methodology that aims to integrate physical and computational components in order to achieve real-time monitoring and control of physical

systems. The seamless integration of computational systems and physical processes depends on the utilization of embedded computers and interconnected networks that incorporate feedback loops. CPS is a new generation of digital systems with two key functional components: advanced connectivity and intelligent data management, analytics, and computational capabilities. Improved connectivity allows for the prompt capture of data from the physical domain, while effective data management and powerful computational skills help to build a dynamic cyberspace environment [90].

Cyber-Physical Systems find extensive utility across various domains, including but not limited to medical devices, traffic control systems, energy conservation initiatives, avionics, and smart structures. WSN are commonly employed in CPS for the purpose of monitoring physical parameters and transmitting data to a central processing unit [91].

CPS, on the other hand, encounters several challenges, such as semantics and concurrency models in computing which can have an impact on the real time performance. Furthermore, the security of CPS is threatened due to internet connectivity. To counteract unauthorized access and potential cyberattacks it is imperative to integrate stringent security measures such as encryption and authentication into the system. Hence, it is required to implement robust security measures, such as encryption and authentication, to protect the system against unauthorized access and cyberattack.

2.13 Characteristics of a WSN

Wireless sensor networks include several key factors that enable them to operate autonomously, adapt to changing environments, and ensure reliable and energy-efficient operation. The architectural composition of a wireless sensor node generally comprises three fundamental components, namely a sensing unit, a processing unit, and a communication unit, as depicted in Figure 2.9. The sensing unit is in charge of detecting and changing physical phenomena such as temperature, pressure, or light into electrical signals. The processing unit is accountable for handling the processing of these signals and performs data analysis and decision-making. The communication unit contains a radio transceiver, which allows the node to transmit and receive wireless signals, as well as other components responsible for data transmission and reception.

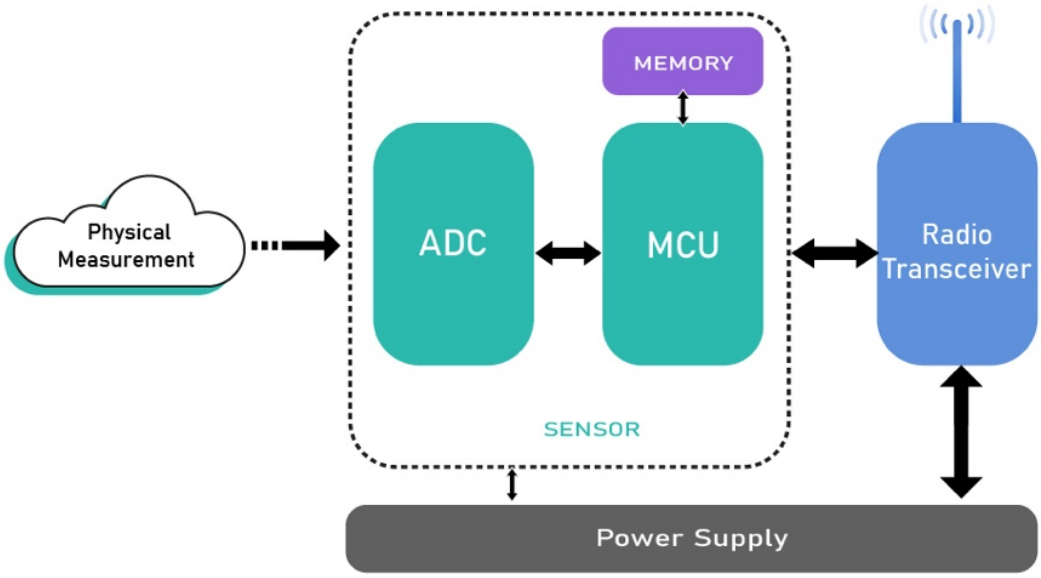

Fig. 2. 9: Typical Architecture of A Wireless Sensor Node [73].

WSNs also possess the following key characteristics:

Resource Constraints: Wireless sensor nodes are constrained in terms of energy, processing power, memory, and communication bandwidth. These limitations affect directly on the design and operation of WSNs. Therefore, the usage of energy efficient protocols and algorithms is essential to optimize power consumption and enhance the network's lifetime. Additionally. Hardware optimizations are necessary to address these limitations respectfully [92].

Energy efficiency: Wireless Sensor Networks (WSNs) usually consist of nodes that are powered by batteries and have limited energy resources; therefore, ensuring energy efficiency becomes a paramount feature for these networks. To extend the network's lifespan, it is imperative to employ protocols and algorithms that are energy efficient. Various methods and techniques can be used to reduce the energy consumption of the nodes while still maintaining the necessary network performance. Examples of such techniques include duty cycling, data aggregation, and energy-aware routing [31].

Fault-tolerance and robustness: Wireless sensor networks may experience issues with individual sensor nodes potentially resulting in a single point of failure. Hence, WSNs must be designed to be fault-tolerant and able to self-repair, reconfigure, and adapt to changing

conditions. The decentralized structure of WSNs empowers them to tolerate the failure of individual nodes without compromising the overall performance of the network. Additionally, the nodes in WSNs are also designed to be robust and resilient to harsh and unpredictable environments. This strategic design guarantees that the network remains operational and dependable under various conditions [93].

Self-organizing and self-configuring nature: WSNs must be able to operate autonomously and adapt to the changing environment. The nodes in a WSN must be able to establish their network topology, configure their communication protocols, and manage their energy resources without any central coordination. This requires the use of distributed algorithms and protocols that enable the nodes to collaborate and coordinate their actions in a decentralized manner [92].

Security and privacy: The protection of sensitive information in collected WSN data is crucial and must be protected from unauthorized access. Therefore, it requires the implementation of robust security measures, such as encryption, authentication, and access control, to guarantee the preservation of data confidentiality. Wireless Sensor Networks require high-level security protection against a diverse range of threats, including eavesdropping, tampering, and malicious attacks [92].

Large number of small and inexpensive nodes: Wireless sensor networks (WSNs) consist of numerous small and cost-effective nodes, which can be deployed in various range of applications and environments. The nodes' compact size and affordability facilitate their large-scale deployment enabling extensive coverage over a wide area. This dense coverage proves beneficial in serving multiple purposes effectively [92].

Autonomous operation without central coordination: WSNs must be able to operate autonomously without any central coordination, allowing them to be deployed in areas that are difficult or impossible to access. This requires the nodes to be able to communicate and collaborate with each other without relying on a centralized controller or coordinator. The autonomous operation of WSNs enables them to operate in various environments, such as underground mines, forests, and oceans [94].

2.14 IoT and WSN Access Technologies

The access technologies for IoT and WSN refer to the diverse methods and protocols that facilitate the connectivity of devices within wireless sensor networks and the internet of things to the internet or other networks. Several access technologies are used in various applications. These include Wi-Fi, Bluetooth Low Energy (BLE), Zigbee, LoRaWAN, cellular networks (such as 3G, 4G, and 5G), RFID, and Near Field Communication (NFC). Each of these technologies presents distinct advantages and disadvantages, contingent upon the specific requirements of the application. For instance, Wi-Fi technology offers high speed connectivity for various devices, whereas BLE (Bluetooth Low Energy) is specifically engineered to facilitate low-power and short-range communication. Zigbee is best suited for industrial and commercial applications, whereas LoRaWAN enables long-range connectivity over large distances. These access technologies can be used in combination to provide the necessary connectivity and functionality for IoT and WSN devices. This section compares ZigBee and LoRaWAN communication protocols, providing a detailed analysis of their respective features, advantages, and limitations.

ZigBee and LoRaWAN are two well-known wireless communication protocols extensively utilized in wireless sensor networks (WSNs). ZigBee is specifically designed for low power and short-range applications operating within the frequency bands of 2.4 GHz or 900 MHz. On the other hand. LoRaWAN caters to long range and low power applications. Operating within sub GHz frequency bands such as 868 MHz in Europe and 915 MHz in North America [95, 96].

ZigBee operates on the IEEE 802.15.4 standard, which is designed for low-rate wireless personal area networks (LR WPANs). This standard outlines the physical and media access control layers for low power wireless communication. The ZigBee Alliance, consisting of over 400 companies, has developed ZigBee as a wireless communication standard for IoT applications. ZigBee solutions are aimed at smart objects and sensors that require low bandwidth and power consumption. These WSNs utilizing ZigBee can effectively monitor various parameters such as temperature, humidity, and light levels. Through research it has been demonstrated that ZigBee based WSNs offer reliable and precise data across diverse applications [**73**].

LoRaWAN on the other hand is a low power wireless communication protocol operating in the sub-GHz frequency bands (868 MHz in Europe and 915 MHz in North America). It is suitable

for WSNs that require long-range, point-to-point communication. LoRaWAN enables monitoring of parameters like temperature, humidity, and pressure. LoRaWAN-based WSNs can provide reliable and accurate data over longer distances. Various studies have demonstrated the capability of LoRaWAN based WSNs to deliver reliable and accurate data in diverse applications [**97**].

ZigBee and LoRaWAN differ in terms of their frequency range and range capabilities; both protocols have their own unique advantages and disadvantages. Choosing the most suitable protocol for a specific application depends on various factors, such as the required range, data rate, and power consumption. In addition, both protocols are suitable for WSNs in various applications, depending on the specific requirements and goals of the application. ZigBee is better suited for short-range, mesh network communication, while LoRaWAN is better suited for long-range, point-to-point communication. Table 2.4 compares the features of both communication protocols in WSNs:

Table 2. 4: Comparison of Features Between Zigbee and LoRaWAN Communication Protocols In WSN.

Feature	ZigBee	LoRaWAN
Frequency	2.4 GHz or 900 MHz	Sub-GHz (868 MHz in Europe, 915 MHz in North America)
Range	Up to 100 meters	Up to 10 kilometers
Power Consumption	Low	Low
Security Features	Encryption, Authentication	Encryption, Authentication
Network Topology	Mesh network	Point-to-point
Suitability for applications	Monitoring various parameters, such as temperature, humidity, and light levels; long-range, point-to-point communication	Monitoring various parameters, such as temperature, humidity, and pressure; long-range, point-to-point communication

The features of ZigBee and LoRaWAN which they are two commonly used wireless communication protocols in WSNs are meticulously compared in the above table. The table provides comprehensive details regarding their frequency bands, range capabilities , power consumption rates , security, network topologies , and suitability for various applications.

CHAPTER 3
RESEARCH METHODOLOGY

In this chapter, we present a proposed key distribution and management scheme for IoT and wireless sensor networks that aims to enhance the security and energy efficiency of the network. The proposed scheme employs elliptic curve cryptography (ECC) and an enhanced version of the Low-Energy Adaptive Clustering Hierarchy (LEACH) routing protocol to provide a lightweight and effective security mechanism for resource-constrained WSNs.

The proposed scheme comprises five stages: clustering algorithm, key initialization algorithm, key distribution, rekeying, and an enhanced version of the LEACH routing protocol. The clustering algorithm divides the nodes into clusters, with each cluster containing an equal number of sensors. This algorithm can take into account factors such as node density, distance between nodes, or other metrics to optimize clustering.

Figures 3.1 and 3.2 illustrate the communication topologies of the proposed scheme. These figures provide a visual representation of the communication processes in the proposed scheme. In addition to intra-cluster communication, where sensor nodes communicate with their respective cluster heads, inter-cluster communication is also demonstrated. The cluster heads of adjacent clusters establish communication with one another by utilizing a multi-hop route, enabling the exchange of information between different clusters without the need for direct communication with the base station or sink node. This approach helps reduce energy consumption and extends the lifetime of the network. The figures demonstrate the importance of clustering in optimizing communication and conserving energy in WSNs.

The key initialization algorithm generates private keys for each node using ECC within a specific range of the curve's field size. The corresponding public keys are then distributed securely to all members of the cluster using the key distribution algorithm, enabling secure communication within the cluster. The rekeying process generates new private keys for each node at specific time intervals or in the event of a group change to maintain the security of the system.

A modified form of the LEACH routing protocol is employed to optimize data transmission and extend the network lifespan, In light of various factors such as the residual energy of the

nodes, the distance to the cluster head, and the communication overhead. This approach enables efficient and effective network operation by reducing the power consumption of the sensor nodes and prolonging the network's lifespan.

Moreover, the proposed scheme improves on existing approaches where sensor nodes forward information to the sink node or base station without clustering or aggregation. The proposed scheme enables single-hop clustering, where each sensor node communicates with its respective cluster head, and the cluster head forwards the aggregated data to the sink node or base station. Additionally, the proposed scheme enables multi-hop clustering, where cluster heads communicate with each other to exchange information, reducing long-distance transmission and saving energy to a large extent.

In summary, this chapter presents a proposed key distribution and management scheme that employs clustering algorithms, key initialization algorithms, key distribution, rekeying, and an enhanced version of the LEACH routing protocol to provide a secure and efficient solution for WSNs. The scheme presents a robust and energy-efficient approach for managing group keys and conserving energy in WSNs.

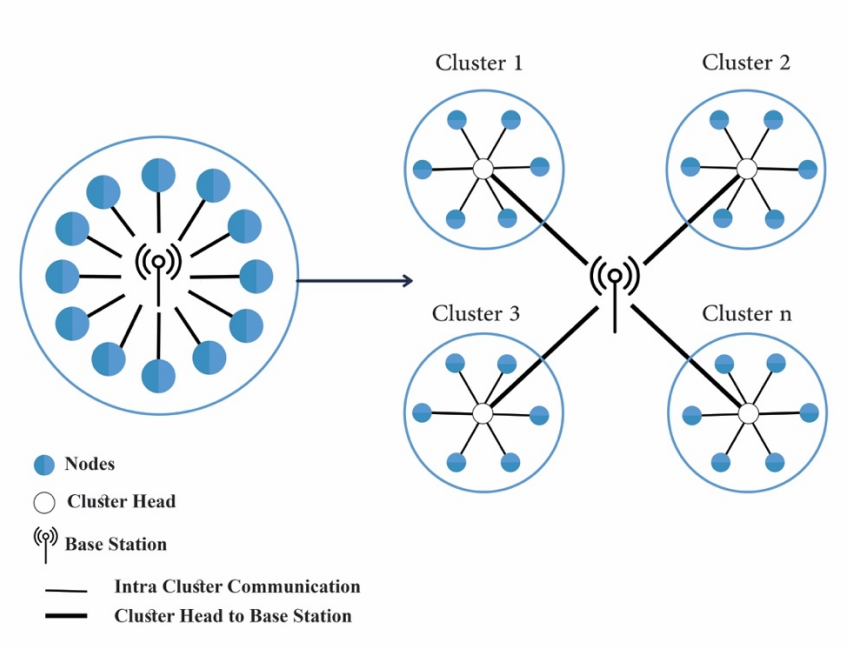

Fig. 3. 1: Topology of A Wireless Sensor Network with Single-Hop Cluster-Based Architecture.

The illustration depicts a Wireless Sensor Network (WSN) with nodes organized into n clusters. To facilitate efficient communication and data administration, sensor nodes within the network are grouped according to their respective clusters. This method of clustering is useful for maximizing the efficiency of the network's lifetime.

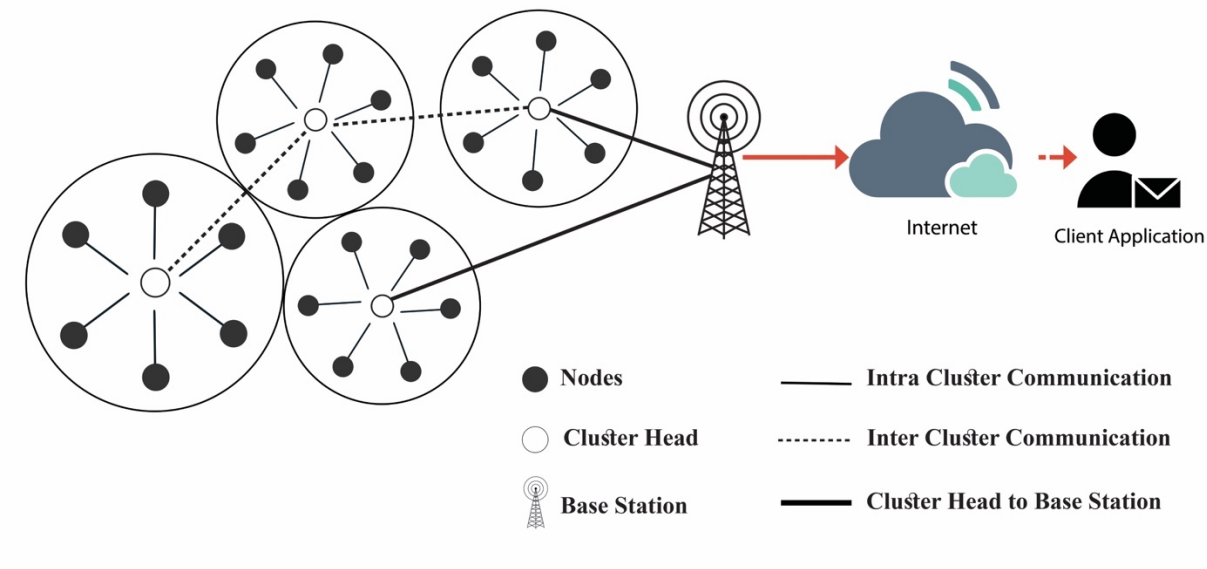

Fig. 3. 2: Topology of A Wireless Sensor Network With Multi-Hop Cluster-Based Architecture[67].

The figure shows how the cluster heads of adjacent clusters communicate with one another through a multi-hop route, enabling the exchange of information between different clusters without having to communicate directly with the base station. This inter-cluster communication is beneficial for reducing long-distance transmission and saving energy to a large extent, as it helps to minimize energy consumption and prolong the lifespan of the network. When the base station receives aggregated data from the cluster heads, it sends it over the internet to user apps for analysis.

3.1 The Initial Key Agreement

The proposed protocol outlines a key-sharing scheme for a group of participants {U1, U2, ..., Un} entities aiming to establish a mutually shared key for secure communication. The participants agree on an elliptic curve E and a point P in E, which is of prime order and sufficiently large to ensure that the Elliptic Curve Discrete Logarithm Problem is computationally infeasible. Each participant Ui holds two pairs of private-public keys (N_i, N_iP) and (X_i, X_iP), which are utilized in the key-sharing process. A designated participant Uc, referred to as the group controller, initiates and manages the key agreement process.

The protocol commences with each participant Ui sharing their private-public key pairs with the other participants in the group. The group controller Uc subsequently broadcasts the initiation of the keying process. Each participant Ui then computes a unique value C_j using their private key and the shared pairs, with the specific calculation being $Cj = Nj(NcP) + Xj(XcP)$, where Nc is the private key of the group controller Uc.

Once all participants have generated their shared keys, the group controller Uc computes a value Nc that represents the common shared key that can be used for secure communication among all participants. This protocol provides a secure means for participants to establish and share a common key without requiring a central authority to oversee the process [98].

Algorithm 1: Initial Key Agreement

1. Each Node U_i in the set $\{U_1, \ldots U_n\}$ share the pairs as (N_i, X_iP) when $i=1,\ldots,n$
2. Group Header U_c broadcast beginning of keying process

$$\{Cj = Nc^{n}_{j=1,j \neq c}$$

3. Each node in the group $U_{i,}$ j =1,…,n, j≠c, use private key and computes $Cj = Nj(NcP) + Xj(XcP)$
4. Group Header U_c computes Nc
5. after all nodes generate their shared keys, the common shared value is Nc($\sum_{r=1,rc}^{n} Nr\ P$

The proposed key-sharing protocol is an extension of the classical Diffie-Hellman key exchange method, which is used over a group of points in an elliptic curve. One can leverage the properties of points in an elliptic curve by considering the sum of points NP + XP for random integers N and X, which can be expressed as the point (N + X)P.

When there are only two users in the group, the key-sharing protocol involves user U1 publishing N_1P in step 1, while user U_2 sends N_2P in step 2. The common shared value (N_1N_2)P is then computed in steps 3 and 4 using the principles of points in an elliptic curve.

The proposed protocol extends the classical Diffie-Hellman key exchange method over the group of points in an elliptic curve by exploiting the properties of points in such curves. This allows the protocol to provide secure communication among multiple users without requiring a central authority to oversee the process.

3.2 Rekeying Process

Rekeying is a procedure that is similar to the group-initiated key, and there are two stages involved in a group rekey. The GH is in charge of handling new group key initialization. Second, a multicast message containing these new keys is distributed to the members of the group. The procedure of creating new keys is similar to the manner used during group establishment; the process starts with selecting members, creates new pairs of keys, and then sends them to the members of the group.

Numerous distributed schemes have been proposed in academic literature to address the rekeying process; however, their suitability for WSNs is limited due to the associated communication overhead and delay [99]. The study conducted by [100] demonstrates that the rekeying procedure in wireless sensor networks (WSNs) requires the utilization of N keys and transmissions, where N denotes the overall number of nodes within the group.

To address the communication burden resulting from rekeying operations in dynamic centralized group communication systems that use key trees, [101] proposes a unidirectional key derivation protocol. This protocol is introduced as a means to mitigate the challenges associated with frequent key updates and their impact on communication efficiency. Tree-based schemes have been employed to achieve improved outcomes in terms of broadcast message size, computational cost, and storage requirements. Nevertheless, when applied to large-scale sensor networks, these approaches still suffer from high communication costs and rekeying delays. Furthermore, a central controller must monitor the status of each node and maintain a logical tree connecting all member nodes, leading to substantial additional workload and elevated overhead.

The paper introduced by [99] proposes a distributed scheme for setting up a group key among a specific group of nodes. However, this scheme has certain drawbacks in terms of scalability and efficiency. When the group size grows, each node in the group has to communicate with other trusted members leading to increased storage cost for each node. Furthermore. This model is only applicable to a single conference key.

There are a number of situations that call for rekeying, such as when a member of the group leaves or joins, or when the shared key expires. This section focuses on the rekeying process that is initiated due to the confidentiality of the shared key. In this case, the group controller

Uc generates a new integer Y and publishes a new pair of public keys ((YN)P, (YX)P). The rekeying process ensures that the shared key is updated, and that the security of the system is maintained. The process is straightforward and enables secure communication within the group by ensuring that all members have access to the updated shared key [98].

Algorithm 2: Rekeying

1. Group Header U_c generates a new integer number Y
2. Share the new pair of public keys *((YN)P, (YX)P)*
3. Group Header U_c broadcast beginning of keying process

   $\{YCj = YNc^{n}_{j=1,\ j \neq c}$
4. Each node in the group $U_{i,}\ j$ =1,…,n, j≠c, use private key and computes $Cj = Nj(NcP) + Xj(XcP)$
5. If a member leaves the group such as: U*i*, then C*i*
6. The group header will transmit a rekeying message

   $\{YCj\}^{n}_{j=1,\ j \neq c,i}$

In the end, when a new member, represented by U_{n+1}, joins the group, they will be provided with two pairs of private and public keys (N_{n+1}, $N_{n+1}P$), (X_{n+1}, $X_{n+1}P$) and will publish their corresponding public keys. The group controller then initiates the rekeying process by sending a message to update the shared key $\{\boldsymbol{C_j'}\}^{\boldsymbol{n+1}}_{\boldsymbol{j=1,j\neq c}}$

$$C_j' = YC_j + YN_{n+1}P, J = 1, \dots, n, n \neq c$$

$$C_{n+1}' = YN_c\left(\sum_{r=1,r\neq c}^{n} N_r\right)P - YX_c(X_{n+1}P)$$

After receiving the rekeying message, each member of the group, denoted by ,*i*=1,…,*n*+1, where $i \neq c$, computes the new shared key $\mathbf{YNc}(\sum_{\boldsymbol{r=1,r\neq c}}^{\boldsymbol{n+1}} \boldsymbol{Nr})\boldsymbol{P}$ using their respective private information (N_i,X_i)This enables each member to update their access to the shared key and securely communicate with other members in the group.

In fact, synchronous rekeying procedures that result in immediate group key rekeying after each request, such as single join and single leave actions, could potentially incur a significant amount of communication overhead. Asynchronous rekeying is able to reduce communication costs by taking advantage of the possibility that new keys for multiple join or leave requests will overlap. This is accomplished by queuing the requests and performing one rekeying for all of them at the same time. According to Lin, Lai, and Lee's findings [**101**], asynchronous rekeying has the potential to alleviate the out-of-sync problem that synchronous rekeying experiences. However, asynchronous rekeying has the drawback of expanding the vulnerability window; however, the security deterioration is typically acceptable in exchange for improved system efficiency. This is because asynchronous rekeying allows for more flexibility in the way that keys are generated.

3.3 Key Distribution and Management using ECC

In this section, the proposed scheme is described as operating on the assumption that the network consists of multiple nodes, all with the same level of authority and characteristics. When a node has data to transmit, it follows a predefined protocol to determine where to send it.

Every node within the network possesses the capability of computing and generating its private key using ECC. These private keys are 256-bit integers within the range of the curve's field size. The proposed scheme makes use of the elliptic curve digital signature algorithm (ECDSA) implemented on the secp256k1 curve, renowned for its notable efficiency and widespread adoption in cryptocurrency systems.

The secp256k1 curve is deliberately constructed in a non-random manner to facilitate efficient computations. In fact, optimized implementations of the ecp256k1 curve can be up to 30% faster than other well-known elliptic curve (EC) curves. The secp256k1 curve is defined by six domain parameters, denoted as a sextuple T = (P, a, b, G, n, h) and it operates over the curve $\boldsymbol{E}(\mathbb{F}_{\boldsymbol{p}})$ specified by equation (1) [102].

In the proposed scheme, the elliptic curve digital signature algorithm (ECDSA) is utilized on the secp256k1 curve, which finds wide application in cryptocurrency systems. The secp256k1 curve is characterized by six domain parameters expressed as a sextuple T = (P, a, b, G, n, and

h) over a curve $\boldsymbol{E}(\mathbb{F}_{\boldsymbol{p}})$ Described in Section 2.9 using Equation (1,2) [102]. These domain parameters play a crucial role in defining the fundamental properties of the elliptic curve and are of utmost importance for the correct implementation and utilization of the ECDSA algorithm.

To generate a new public key within the proposed key distribution and management strategy, every individual sensor node performs a scalar multiplication operation between its private key and the reference point on the curve. This computation yields a new point on the curve, which serves as the node's public key.

The scheme incorporates a distributed key agreement protocol that builds upon the pioneering work of Diffie-Hellman. Furthermore, advancements in the routing protocol have been documented in reference [29]. The protocol, outlined in Algorithm 3, facilitates a secure consensus among nodes to establish a mutually agreed-upon key within a distributed framework [32].

Algorithm 3: Cyclic Elliptic Curve Cryptography (ECC) Key Generation and Distribution Algorithm

1. Node N_g broadcast beginning of keying process
2. Initialize $P_{i-1} \leftarrow G$
3. Choose node j to generate its shared key P_j
4. While $i <= N - 1$ do:
5. If $i \neq j$
6. Send public key P_{i-1} to node i
7. Node j computes new: $P_{i-1} \leftarrow r_i P_{i-1}$
8. End
9. Send public key P_{i-1} to node j
10. node j computes it shared key as: $P_j = r_j . P_{i-1}$
11. repeat step 2 to 10 until all nodes generate their shared keys

Steps (1–2). Initially, the node GHi ($\boldsymbol{N_g}$) creates a message called M_1 and sends it to all members of the network. The message requests that all transmissions be stopped for a period of time that is determined by the key initialization protocol.

Steps (3–8). The process of key exchange is initiated by the group head, who sends a point $\boldsymbol{P_{i-1}}$ to node I. This point is the generator point *G* on an elliptic curve. The group head then performs a cyclic public key exchange with all other nodes in the network, except for node j. Each node receives a point $\boldsymbol{P_{i-1}}$ from the group head and uses its private key to compute a new point based on elliptic curve scalar multiplication using the formula (2), where $\boldsymbol{r_i}$ is the private key of node *i*, and $\boldsymbol{P_{i-1}}$ is the public key computed by the previous node $\boldsymbol{i-1}$. Therefore, all nodes in the network are involved in the key exchange process.

$$P = \prod_{i=1, i\neq j}^{N-1} r_i P_{i-1} \quad (4)$$

Steps (9–10). When the cyclic exchange is finished, node $\boldsymbol{j}$ receives the most recently computed public key $\boldsymbol{P}$ from $\boldsymbol{N_g}$. It then employs it to compute its own shared key, as seen by Equation (5).

$$P_j = r_j . P \quad (5)$$

Step (11). During this particular stage, the ongoing procedure will continue until every node within the network successfully calculates its shared key, as depicted in Figure 3.3.

In a network containing five nodes, identified as GH, N1, N2, N3, and N4, each with corresponding private keys $\boldsymbol{a, b, c, d, and\ e}$, respectively, the group head node GH initiates the process of generating and distributing shared keys between nodes N1 and N2, as depicted in Figure 3.3. To generate the shared key $\boldsymbol{P_{n1}}$ for node N1, the private keys of GH, N2, N3, and N4 are multiplied together, and the resulting product is scalar multiplied by the private key of node N1, with the generator point G on the elliptic curve E. This can be expressed mathematically as :

$$P_{n1} = \left(r_{gh} . r_{n2} . r_{n3} . r_{n4} . G\right) . r_{n1}$$

where r_{gh} is the private key of node GH, r_{n1} is the private key of node N1, and "." represents scalar multiplication.

Similarly, to generate the shared key P_{n2} for node N2, the private key of node N2 is used in place of the private key of node N1 in the above equation. This can be expressed as:

$$P_{n2} = \left(r_{gh}.r_{n1}.r_{n3}.r_{n4}.G\right).r_{n2}.$$

These equations are used to compute and distribute shared keys between nodes N1 and N2, utilizing a combination of private keys and scalar multiplication on an elliptic curve.

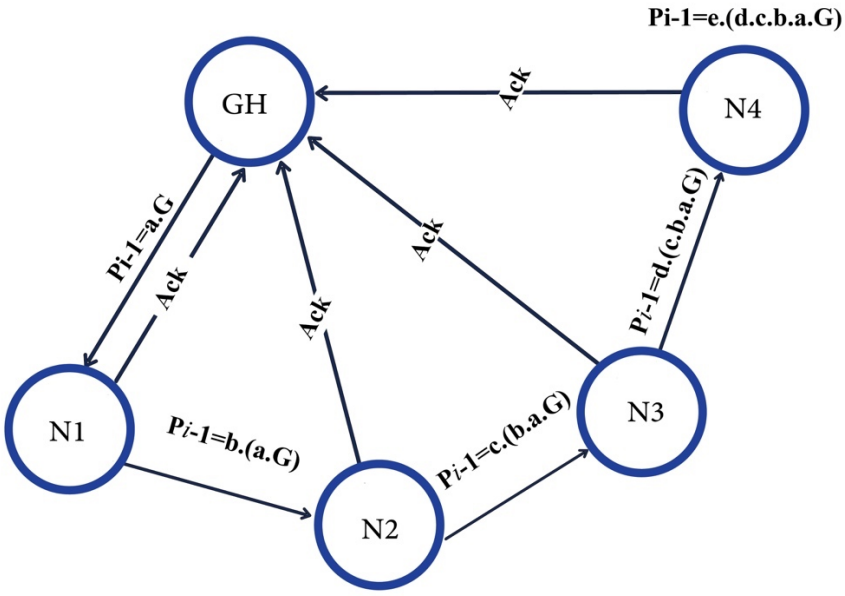

Fig. 3. 3: Key Generation and Distribution Among Nodes in A Network of Five Nodes.

The figure shows all the steps of generating and distributing keys between nodes such as N1 and N2 in a network containing five nodes. The group's leader node GH starts the process by multiplying the private keys of nodes GH, N2, N3, and N4 together and scalar multiplying the result by node N1's private key, using the generator point G on the elliptic curve E. This produces the shared key Pn1 for node N1. Similarly, the private key of node N2 is used in place of the private key of node N1 to produce the shared key Pn2 for node N2. A combination of private keys and scalar multiplication on an elliptic curve is used in the procedure, resulting in the secure exchange of shared keys between nodes N1 and N2.

Our suggested approach for addressing the problem of group rekeying in wireless sensor networks (WSNs) comprises a set of procedures outlined in Table Algorithm 4 [32]. These procedures can be executed at various nodes within the network, either collectively or independently. When the membership of a group changes, the updated group key must be distributed promptly and reliably to the remaining nodes in the network while maintaining security and performance. To prevent potential intruders from gaining an unfair advantage, we propose

a group rekeying technique that is independent and generates a new set of keys. Furthermore, we use a modified version of the LEACH routing protocol to update the routing table.

Algorithm 4 : EC Rekeying

1 Start
2 Initialize all the buffers in the nodes, i.e., P_{mid}- =-1, NN, P, G and x_i
3 Select a node to initiate the keying process
4 Generate new message with fields TYP=0, NC=0, P_x=G_x, P_y=G_y,S_{id}
5 Set the new message Id $M_{id} = P_{mid} + 1$
6 Compute the new points using EC scalar multiplication
7 Increment NC by 1 and choose the next destination node D_{id} sequentially.
8 While D_{id} are not completely used up, do the following:
9 Route the message over the network and performs the following at each new hop
10 If P_{mid} is equals to M_{id} (i.e., old message hoping over)
11 Route the message over the network to another node
12 If P_{mid} is not equals to M_{id} (i.e., new message arrival)
13 Update the points on the curve P_x and P_y such that $P_x = x_i P_{x_{old}}$and $P_y = x_i P_{y_{old}}$
14 Increment NC by 1
15 Check if NC is equals to NN-1 (i.e., destination node after visiting all other nodes)
16 Return the shared key for the node i as $S_{key} = P_x \vee P_y$
17 Iterate steps 4 to 19
18 If NC is not equals to NN-1 (i.e., new message arrives at non-destination node)
19 Iterate steps 8 to 19
20 stop

The above algorithm is used as a rekeying process using EC in a wireless sensor network (WSN) when a node leaves the group. This algorithm describes a key establishment protocol. Here are the step-by-step explanations:

- Initialize all the buffers in the nodes, i.e., $\boldsymbol{P_{mid}}$ =-1, NN, P, $\boldsymbol{G}$ and $\boldsymbol{x_i}$ This step sets up the initial state of the network nodes for the key establishment protocol. The buffers are initialized with appropriate values.
- Choose a node to start keying: A node is chosen as the starting point for the protocol.

- Generate new message with fields TYP=0, NC=0, $\boldsymbol{P_x}$=$\boldsymbol{G_x}$, $\boldsymbol{P_y}$=$\boldsymbol{G_y}$,$\boldsymbol{S_{id}}$ A new message is generated with several fields, including the message type (TYP), node count (NC), coordinates of a point on the elliptic curve ($\boldsymbol{P_x}$ and $\boldsymbol{P_y}$), and a sender ID (S_id).
- Set the new message Id $\boldsymbol{M_{id}}$=$\boldsymbol{P_{mid}}$ +1: A unique message ID is assigned to the new message, which is one greater than the previous message ID ($\boldsymbol{P_{mid}}$).
- Calculate the new points by performing EC scalar multiplication: The elliptic curve scalar multiplication operation is used to generate new points on the curve based on the sender's private key and the generator point (G).
- Increment NC by 1 and choose the next destination node $\boldsymbol{D_{id}}$ sequentially: The node count (NC) is incremented, and the next destination node is chosen sequentially.
- While $\boldsymbol{D_{id}}$are not completely used up, do the following: This step initiates a loop that runs until all destination nodes have been visited.
- Route the message through the network and carry out the subsequent actions at each new hop: The message is routed over the network to the next node, and the following steps are performed at each new hop.
- If $\boldsymbol{M_{id}}$ is equal $\boldsymbol{P_{mid}}$ (i.e., old message hopping over): If the current message ID ($\boldsymbol{P_{mid}}$) is equal to the previous message ID ($\boldsymbol{M_{id}}$), it means that the message has already visited this node, and the message is routed to another node.
- If $\boldsymbol{M_{id}}$ not equal $\boldsymbol{P_{mid}}$ (i.e., new message arrival): If the current message ID ($\boldsymbol{P_{mid}}$) is not equal to the previous message ID ($\boldsymbol{M_{id}}$), it means that a new message has arrived at this node, and the points on the curve ($\boldsymbol{P_x} \wedge \boldsymbol{P_y}$) are updated.
- Update the points on the curve $\boldsymbol{P_x} \wedge \boldsymbol{P_y}$ such that $\boldsymbol{P_x = x_i P_{x_{old}}}$and $\boldsymbol{P_y = x_i P_{y_{old}}}$): The points on the curve are updated using elliptic curve scalar multiplication.
- Increment NC by 1: The node count is incremented.
- Check if NC is equals to NN-1 (i.e., destination node after visiting all other nodes): If the node count is equal to the number of nodes minus one, it means that the current node is the destination node.
- Return the shared key for the node i as S_key=P_x or P_y: The shared key for the current node is computed as either the x-coordinate or y-coordinate of the updated point on the curve.
- Iterate through steps 4 to 19: The protocol is repeated from step 4 to step 19 for the next message.

- If NC is not equals to NN-1 (i.e., new message arrives at non-destination node): If the node count is not equal to the number of nodes minus one, it means that a new message has arrived at a non-destination node.
- Iterate through steps 8 to 19: The loop continues from step 8 to step 19 for the next hop.
- Stop: The protocol ends when all messages have been sent and all.

The algorithm describes a key establishment protocol that utilizes elliptic curve cryptography over a network. The protocol involves generating a new message with appropriate fields, computing new points on the curve using elliptic curve scalar multiplication, and routing the message to the next node sequentially. At each hop, the points on the curve are updated, and the node count is incremented. If the node count reaches the number of nodes minus one, the current node is the destination node, and the shared key for that node is computed. The protocol is repeated for each message until all messages have been sent.

3.4 Enhanced LEACH Routing Protocol

LEACH is a well-known protocol for WSNs that provides significant energy savings. However, there are several shortcomings associated with the protocol. One major issue is that the selection of a new cluster head is not based on the residual energy of the node. This can result in the loss of aggregate data for that cluster header if it dies midway through the round, and no data can be transmitted or received successfully until the next round. One of the primary concerns regarding this protocol is the lack of consideration given to the residual energy of the node during the selection process for a new cluster head. Additionally, the selection of a new cluster head only occurs once a complete round has been completed. In the event that a selected cluster head fails during a round, all the aggregated data associated with that cluster head is lost, leading to the inability to successfully transmit or receive any data until the subsequent round when a new cluster head is appointed.

LEACH possesses another constraint whereby the selection process for cluster heads involves either random selection or relies only on energy levels; disregarding crucial factors such as the distance to the sink and the package size play a determining role in the energy consumption for transmitting the message.

The most qualified node may not always be the one with the highest energy, particularly when situated at the greatest distance from the base station. Additionally, to summarize, the criteria for replacing a cluster head (CH) often involve ensuring it maintains a minimum energy level to efficiently collect and transmit data from leaf nodes, light nodes, or sensor nodes through regular links while considering power consumption requirements. An additional factor that could initiate a request for replacement is the identification of an Advanced Persistent Threat (APT) targeting the cluster head . Detection of such a threat becomes more feasible by leveraging Distributed Ledger Technologies (DLTs), which entail a decentralized database managed by a distributed group of users across multiple nodes. In conjunction with DLTs, smart contracts stored on a blockchain can be employed. Smart contracts are programmable codes that execute automatically when predetermined conditions are satisfied. [103]. In light of these circumstances, an alert mode is activated, initiating urgent rekeying operations.

To address these issues, we proposed improvements to the original LEACH methodology, which are detailed below:

The cluster header is responsible for calculating the amount of energy needed to transmit a packet $\boldsymbol{E_{tx}}$. If the required energy to transmit a packet, $\boldsymbol{E_{tx}}$, is less than or equal to the node's residual energy ($\boldsymbol{E_{residual}}$, the cluster header will resign its status and a new cluster header will be selected. The data aggregated within the previous cluster header necessitates relocation to the newly appointed cluster header. The computation of the requisite transmit energy for a packet of size Packet Length (7) can be accomplished using Equations (6) and (7).

The cluster header calculates the energy needed to send a packet before it is sent, $\boldsymbol{E_{tx}}$. If the required energy to transmit a packet, $\boldsymbol{E_{tx}}$, is less than or equal to the residual energy $\boldsymbol{E_{residual}}$ of the node (i.e., $\boldsymbol{E_{tx} \leq E_{residual}}$), consequently, the cluster header will relinquish its position, leading to the selection of a new cluster header. The accumulated data residing in the former cluster header must be seamlessly transferred to the newly elected cluster header. To ascertain the essential, transmit energy for a packet with a size of Packet Length, one can compute it using Equations (6) and (7) as per the established methodology.

$$E_{tx} = S_{Etx} \times PacketLength + S_{Efs} \times d^2 \quad (6)$$

$$E_{tx} = S_{Etx} \times PacketLength + S_{Emp} \times d^4 \quad (7)$$

where S_{Etx} is the base station transmitter energy parameter and S_{Emp} and S_{Efs} are the energy parameters for the radio transmitter type in the multipath and free space models, respectively. The parameter d is the distance between the node and the base station S.

Equation (8) calculates the Euclidean distance, denoted as d_i, between node *i* and the base station S. The coordinates of node i are given by (x_i, y_i), and the coordinates of the base station S are given by (x_s, y_s). The distance energy-factor E_{ed} which is used to determine the cluster header, can be calculated using Equation (9). The formula for Euclidean distance between node *i* and the base station S, using Equation (8), is:

$$d_i = \sqrt[2]{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{8}$$

Where Xi, Yi and Xs, Ys are the two-dimensional coordinates of either the node or the base station. The node that has been determined to have the greatest value is always selected to serve as the cluster header. In contrast to the LEACH protocol's use of only nodes with the highest energy levels or random selection procedures, this ensures that the distance between the node and the base station is also taken into consideration during the selection process. This replaces the use of nodes with the highest energy levels only.

$$E_{ed} = E_{residual} - E_{tx} \times \frac{1}{d^2} \tag{9}$$

Furthermore, in the cluster head selection process, only nodes meeting the criterion of having residual energy greater than their transit energy are taken into account as potential candidates for the cluster head position.

CHAPTER 4
SIMULATIONS RESULTS AND PROPOSED MODEL EVALUATION

This study presents a model of key distribution and management to bolster the security and energy efficiency of IoT and wireless sensor networks (WSNs). The proposed model employs elliptic curve cryptography (ECC) and an enhanced version of the Low-Energy Adaptive Clustering Hierarchy (LEACH) routing protocol to provide a lightweight and effective security mechanism for resource-constrained WSNs.

To validate the proposed model, various parameters were simulated using MATLAB considering different scenarios. The obtained simulation results were then compared with those of previous models for key distribution, key exchange, and routing protocols, including the work of Heinzelman et al. [31] and Saini and Sharma [104].

In a sensing field of 100 m × 100 m with random spacing between nodes, we simulated a wireless sensor network of 50 and 100 nodes, respectively. The simulation settings used in the test bed are shown in Table 4.1 [32]. The table lists all of the parameters and their corresponding values utilized in the simulation. In addition to the sensor nodes, there is a single base station in the middle of the network that is responsible for receiving information from those sensors.

Initially, sensor nodes will transmit their identification number as well as information regarding their remaining energy. The estimation of distances between adjacent nodes can be achieved by utilizing the received signal's strength. Within the presented context, the assumption is made that the base station deploys an omni-directional antenna characterized by zero gain and no system loss. Initial energy capacities for all nodes are set to 0.005 joules, while the base station has an unlimited power supply.

Table 4. 1: Parameters Used for Proposed Improved LEACH Protocol.

Definition	**Average Value**
Size Of Testbed (# Nodes)	50,100 homogenous
Number Of Base Station (Bs)	1
Initial Energy for Each Sensor	0.005 J/battery
Radio Circuitry Energy Dissipation, Eelec	50 nj/bit
Energy Dissipation of Amplifier In Free-Space, Efs	10 pj/bit
Energy Dissipation of Amplifier in Multipath, Emp	0.0013 pj/bit
Energy Consumption for Data Aggregation, Eda	5 nj/bit
Threshold distance (d_0)	n m
Global Testbed Area	n*n
Local Area (Cluster Size)	n/nc
Time	10 rounds
Packet Size	400 bits
Message Size	328 bits
Encryption Key Length	256 bits

The clustering formation process encompasses multiple stages aimed at determining the optimal number of clusters, cluster head (CH) nodes, and member nodes associated with each cluster. In our simulation, we employ a formula to divide the nodes into a predetermined number of clusters denoted as 'n'. To evaluate the similarity among nodes and facilitate clustering, we leverage the respective energy consumptions as similarity measures.

To determine an appropriate number of clusters for our simulation, we refer to the research conducted by [105]. Their findings suggest that in networks consisting of 100 nodes, an optimal range of 3 to 5 Clusters tend to yield great outcomes. Guided by this insight, we incrementally increased the number of clusters from a single cluster to a total of 4 clusters. When the number of clusters is below 3, the size of each cluster becomes large, resulting in increased energy expenditure for non-cluster head nodes to communicate with the cluster head. Consequently, the network lifetime is shorter in such scenarios. On the other hand, LEACH demonstrates that the maximum network lifetime is achieved when the number of clusters is set to 5.

Conversely, when the number of clusters exceeds 6, each cluster contains a small number of members, leading to an increased frequency of packet transmission from nodes to the cluster

head. As a result, both the sensor nodes and the cluster heads consume more energy to handle the increased data load. Consequently, the network lifetime significantly decreases as the number of clusters increases. From the observations made, it becomes apparent that a trade-off exists between the number of clusters and energy consumption. The optimal number of clusters lies between 3 and 5, striking a balance between efficient energy utilization and network lifetime.

The simulation was executed over a duration of 1500 simulated rounds, during which the nodes were randomly distributed across the network and evenly allocated to each cluster. No explicit constraints or limitations were imposed on the spatial proximity between the nodes and their respective cluster heads. In this regard, the nodes were positioned randomly and uniformly within each cluster, without enforcing any specific restrictions on the inter-node distances or the distances to their corresponding cluster heads. A visual representation of the network topology is provided in Figure 4.1, illustrating a wireless sensor network comprising 100 nodes divided into four distinct clusters using MATLAB.

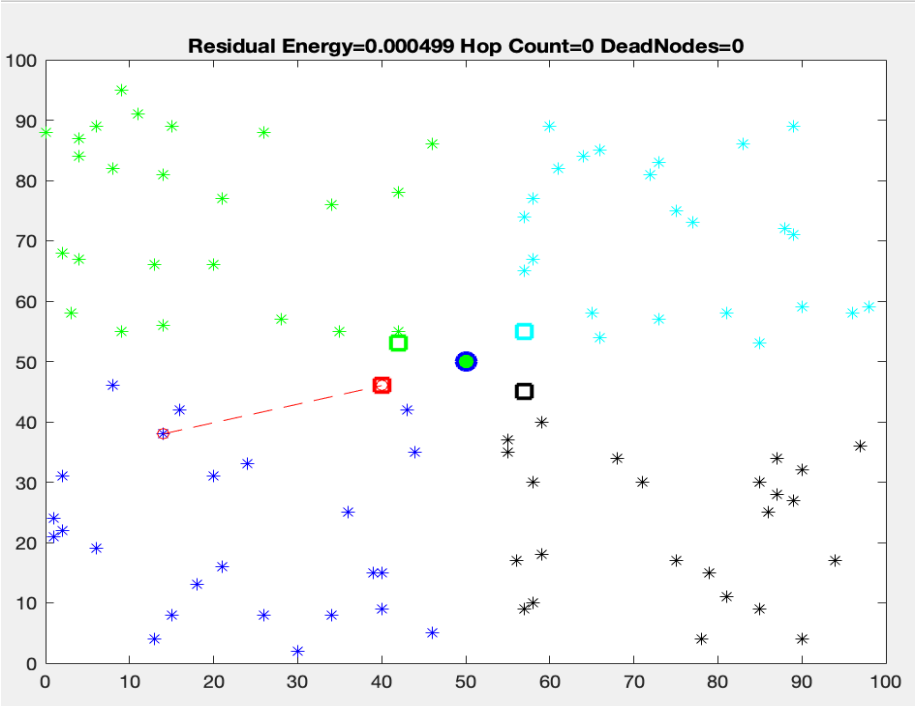

Fig. 4. 1: Wireless Sensor Network Topology for a Network Consisting of 100 Nodes.

The figure depicts the topology of a wireless sensor network that consists of 100 nodes, which have been divided into 4 clusters using MATLAB. The clustering has been performed using both the Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm and its enhanced version to select the cluster heads.

While we employed a Minimum Viable Device (MVD) configuration for the sensor nodes in our sensor network with an ARM processor, chipset Qualcomm MSM8974 Snapdragon 800, CPU quad-core 2.3 GHz Krait 400, storage capacity 16 or 32 GB, and RAM 2 GB [98]. We used the cycle-accurate micro-architectural simulator gem5 to simulate the MVD as shown in figure 4.2.

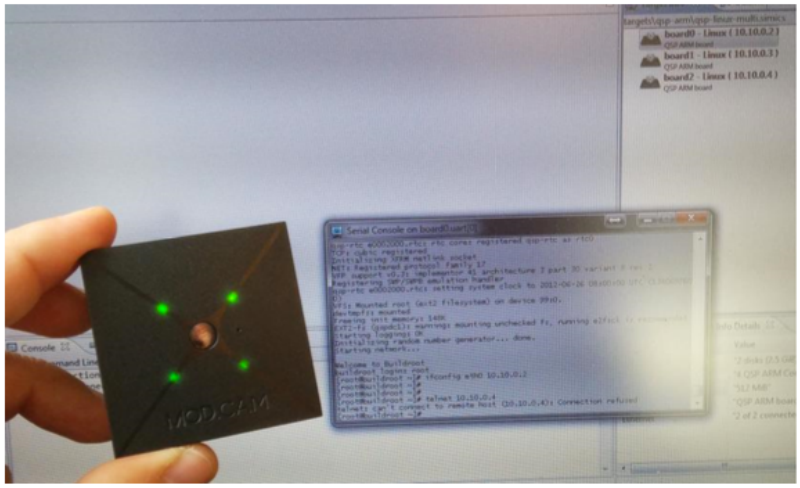

Fig. 4. 2: ARM-Based MVD Node For Sensor Network [98].

The figure illustrates an ARM-based MVD (Multimedia Video Display) node designed for a sensor network. It showcases the key features of the node, including the utilization of the Qualcomm Snapdragon 800 chipset, which incorporates a Quad-core 2.3 GHz Krait 400 CPU, 2GB RAM, and offers a storage capacity of either 16GB or 32GB.

The network stability period refers to the duration from the initiation of network operation until the death of the first node within the network. The instability period is defined as the time span from the death of the first node until the death of the last node in the network. It is worth noting that a network with higher stability is associated with a longer lifetime.

The simulation results presented in Figure 4.3 demonstrate that the difference between both protocols in term of utilizing the energy consumption which affect the network stability and longevity. When compared to the original LEACH protocol, the stability period and node lifetime of the modified version were shown to be significantly higher. This improvement can be attributed to the modified protocol's use of residual energy, which prolongs the life of nodes and results in a slower rate of death. In contrast, the original LEACH protocol exhibited a faster rate of node death after round five and had a final dead node in round nine, leading to a shorter network’s lifetime.

In addition, it was discovered that the suggested protocol has a last dead node that lasts up to 10 rounds, indicating that it exceeds the original LEACH in terms of network stability and lifespan. This enhancement is mostly due to the proposed protocol's enhanced threshold condition, which leads to the network's increased stability.

These findings highlight the significance of optimizing network performance in wireless sensor networks, particularly in terms of increasing the network lifetime and network survivability. By employing effective routing metrics, such as a combination of the shortest path algorithm and network lifetime extension, along with strategies that make use of residual energy, network stability and longevity can be significantly improved.

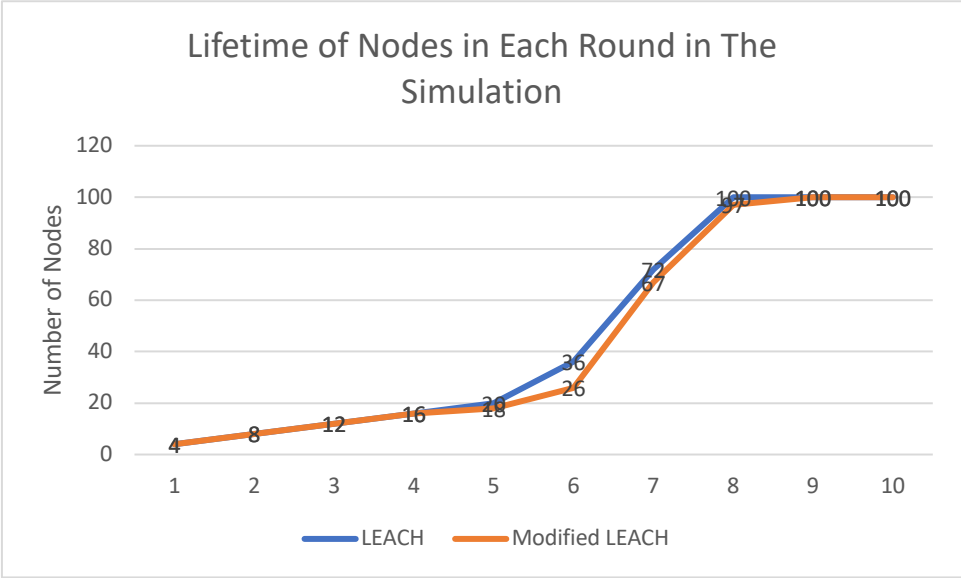

Fig. 4. 3: Comparison of The Number of Rounds and Dead Nodes in A Network Of 100 Nodes.

The x-axis represents the number of rounds, and the y-axis represents the number of operational nodes. The graph shows how the number of functioning nodes changes over time for both the original and modified LEACH protocols. Up to 100 rounds, both protocols maintain a similar number of working nodes. Nonetheless, the revised LEACH protocol sustains a greater count of operational nodes when contrasted with the original LEACH protocol, particularly in the later rounds. This finding supports the conclusion that the modified LEACH protocol outperforms the original LEACH protocol in terms of network stability and longevity, as it is able to prolong the life of nodes and maintain a higher number of operational nodes over time.

The computational load and complexity of key management in wireless sensor networks are significantly impacted by the available resources of the nodes and the dynamic nature of the network architecture. The key management procedures required to share a session key with other entities may be too heavy for sensor nodes, which are often restricted devices with limited processing capacity.

The time complexity for establishing and recovering the group key by each group member is stated in our proposal method, where computing the computational comparison of the proposed protocol and LEACH protocol is shown in Figure 4.4. The result shows a significant decrease in time consumption in each round. This is primarily due to the fact that the distance factor and remaining energy are considered while selecting cluster heads. In the modified LEACH protocol, nodes closer to the sink are chosen as head nodes more frequently than nodes further apart. This means that the overall time required for each round is reduced. In the LEACH protocol, the cluster head is rotated among the nodes to prevent a single node from consuming an excessive amount of energy. LEACH-C is an additional variant of LEACH. In this approach, cluster creation and cluster head selection are centralized and carried out by the base station after receiving all information from the sensor nodes in each round. LEACH presupposes that all nodes are accessible to one another and that all nodes are eligible to serve as cluster heads.

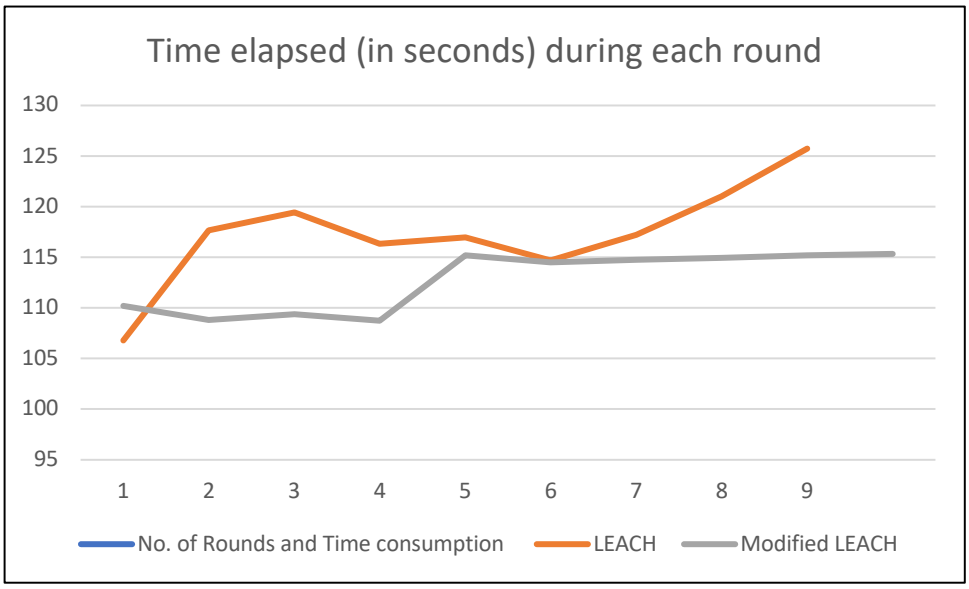

Fig. 4. 4: The Average Round Time for a Network of 100 Nodes.

The figure showcases the overall time consumption per round in a network comprising 100 nodes. It provides an analysis of the time required for various operations within each round of the network. Additionally, the figure highlights the time consumption difference between a proposed protocol and LEACH, allowing for a comparative assessment of their respective efficiencies.

A sensor node in a wireless sensor network collaborates with neighbouring nodes to facilitate data transmission. This is referred to as the multi-hop relay. When such a network is utilized for a particular application, sensor nodes are typically dispersed in a certain area to collect data. The data that has been gathered is transmitted to a central station using a multi-hop relay

technique for further analysis or remote monitoring and control. Consequently, a routing mechanism is required to ensure message delivery.

A robust routing system would be able to dynamically alter the transmission path in accordance with the situation, even if there were node failures. Finding the appropriate routing path for transmitting collected data from a source node to a sink node is an essential challenge that must be overcome in wireless sensor networks. The best path could be determined in a number of different ways due to the fact that wireless sensor networks are not identical to one another. A few examples of potential elements, also known as metrics, to take into account are energy dissipation, radio coverage range, and the number of nodes that are utilized when determining the ideal route.

These parameters may be considered individually or in any combination. As the number of sensor nodes increases, so does the importance of a management strategy to choose among those measurements. Figure 4.5 depicts the probability threshold of reaching cluster head. It is clear that the highest number of hops required to reach cluster head is for the original LEACH protocol. As the network grows, the number of hops will also increase.

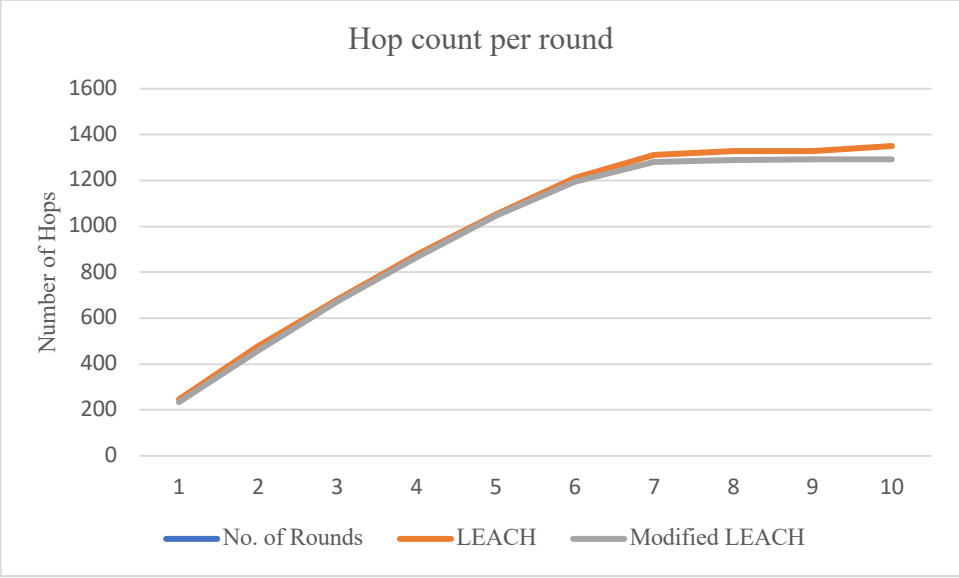

Fig. 4. 5: Hop Count per Round in a Network Comprising 100 Nodes.

The figure presents the overall hop count per round in a network consisting of 100 nodes. It provides insights into the number of hops required for data transmission or communication between nodes within each round of the network. The hop count represents the number of intermediate nodes that a message or data packet must traverse to reach its destination.

Figure 4.6 depicts a performance comparison of three different protocols: the proposed enhanced LEACH protocol (E-LEACH), the original LEACH protocol, and an additional version of the improved LEACH protocol known as E-DEEC [104, 106]. The simulation for this analysis used the same node parameters as shown in Table 4.1, ensuring a fair and consistent comparison. One hundred sensors were randomly deployed across the designated field and remained stationary throughout the experiment, to simulate realistic sensor behaviour. The simulation was conducted over 1500 cycles, allowing for a thorough evaluation of the performance of each protocol.

The number of nodes that were found to be dead after each round was the major parameter examined. By tracking this metric, it was possible to evaluate the longevity and stability of the network under the three different protocols. Figure 4.6 graphically presents the results of this analysis, providing insights into the comparative performance of E-LEACH, the original LEACH protocol, and E-DEEC.

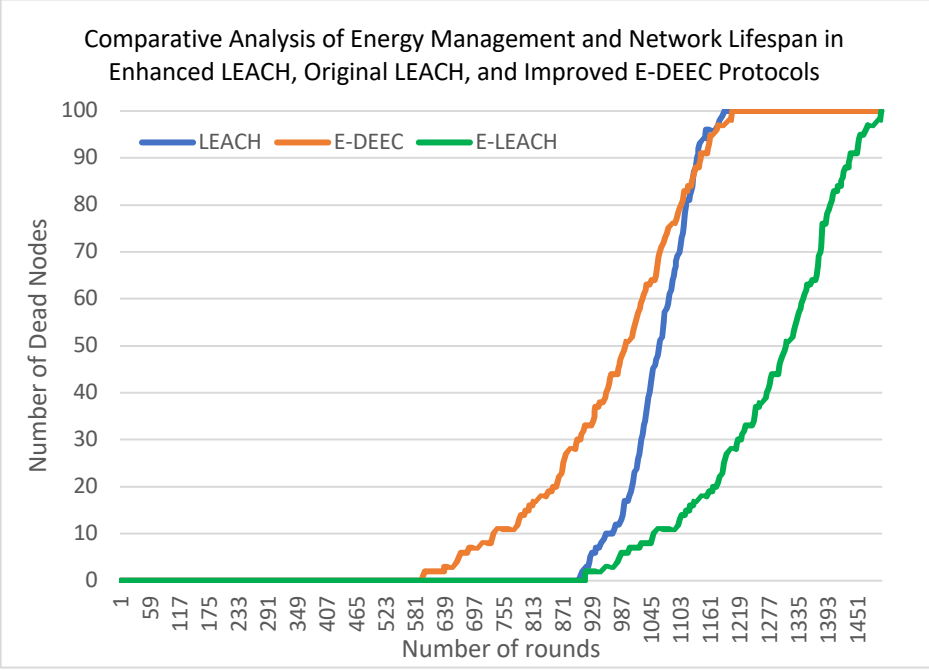

Fig. 4. 6: Comparison of Enhanced LEACH, Original LEACH, And Improved E-DEEC Protocols with a Focus on Energy Management and Network Lifespan.

The graphic compares three protocols in terms of energy management and network lifespan in a WSNs: Enhanced LEACH, Original LEACH, and Improved E-DEEC. It displays a graphical depiction of the performance data associated with each protocol, highlighting their respective energy efficiency and network operating time before battery depletion. This comparison provides a more in-depth insight of each protocol's strengths and limitations in terms of energy utilization and overall network lifetime.

The simulation findings show that the three protocols perform significantly differently in terms of the occurrence of dead nodes and total network longevity. Dead nodes began to arise in the original LEACH protocol at 600 rounds, and by approximately 1100 rounds, all nodes had become inactive. This indicates a relatively short network lifespan and challenges in maintaining the network's operational state over an extended period.

As an enhanced version of the original LEACH protocol, the E-DEEC protocol shows some improvement. It delyaed the first occurrence of dead nodes by about 900 rounds, indicating greater energy management and slightly longer network longevity than the original protocol. However, similar to the original LEACH, all nodes eventually turned inactive around the same time, indicating constraints in terms of network activity sustainability.

The researchers were able to successfully decrease the number of dead nodes that occurred in the network. These dead nodes only started appearing after about 904 rounds. It is worth mentioning that even after 1496 rounds there were still active nodes in the network. This discovery indicates better energy management and a considerably extended lifespan for the network when compared to both the original LEACH and E DEEC protocol.

The results suggest that the enhanced LEACH protocol has better performance by delaying the occurrence of dead nodes and extending the networks lifespan. These findings emphasize the potential of the enhanced protocol to improve energy efficiency and overall network sustainability. Further analysis and statistical evaluation can offer deeper insights into the performance variations and help optimize the protocols for more efficient wireless sensor network operation [32].

The comprehensive findings clearly demonstrate that when it comes to managing energy levels and extending network lifespan, the enhanced LEACH protocol outperforms both the original LEACH protocol as well as the improved E-DEEC protocol. These significant results bear important implications for designing IoT routing protocols, underscoring the need for effective strategies in managing energy levels to prolong wireless sensor networks' lifespans.

The results also demonstrate that the proposed protocol extends the lifespan of networks, providing additional benefits beyond security and energy efficiency. These findings have important implications for the development and deployment of secure and energy-efficient communication protocols in resource-constrained IoT and WSNs, highlighting the potential of the

proposed protocol to significantly increase the lifespan of networks while maintaining security and energy efficiency.

4.1 Evaluation of the Protocol

The evaluation setup involves configuring the physical nodes with a minimum viable device (MVD) and using the gem5 [107] micro-architectural simulator to emulate the system. Assigning the most powerful node as the Uc ensures that the increased computational burden due to a growing number of users does not adversely impact the protocol's overall performance.

The proposed group key agreement protocol demonstrates several advantages over the ING and BD protocols in terms of communication costs, efficiency, and scalability. Our protocol requires the least amount of bandwidth due to the minimal number of messages being sent, as shown in Table 4.2. Each member sends and receives only one message, reducing the burden on the communication infrastructure. This is especially important for sensor networks, where resources are constrained, and efficient use of available bandwidth is crucial.

Table 4. 2: Comparative Analysis of Protocols.

Parameter\ Protocol	ING	BD	Proposed Protocol
Rounds	*n-1*	2	2
Number of Messages	*n(n-1)*	*2n*	*(n-1) 1 by* U_c
Transmitted Messages	*n-1*	2	1 1 U_c
Received Messages	*n-1*	*n+1*	1
Modular Exponentiation by U_i	*n*	*n-1*	-
Modular Multiplication by U_i	-	-	4 by U_i $2(n-1)+1$ U_c

The provided table compares the ING, BD, and proposed protocols based on different parameters, including rounds, transmitted, and received messages, and modular exponentiation and multiplication by Ui.

Moreover, the proposed protocol leverages elliptic curve cryptography (ECC) to offer significantly faster computation times than legacy methods. ECC allows the protocol to provide the same level of security with shorter key sizes, which results in lower computational overhead.

As previously mentioned, the gem5 [107] simulator was utilized to successfully emulate the MVD protocol. To ensure optimal performance within the protocol, it is crucial to assign the most powerful node to the Uc (User with computational-intensive tasks) role. As the number of users increases, the computational workload grows, impacting other protocol components. Users U1 to Uj (excluding the controller) consistently engage in four modular multiplications and a summation operation involving a number (Cj) whose size depends on the number of users. Analysis in Figure 4.7 indicates that the node assigned to the Uc role dedicates more time to operations when dealing with larger integer sizes. Evaluations were conducted using both native and non-native datatypes, revealing that the Uc node experiences greater delays with non-native datatypes. Nonetheless, the protocol exhibited commendable performance even when a substantial number of nodes, such as 1024, were present.

The gem5 simulator proved valuable for emulating the MVD protocol. Assigning the most powerful node to the Uc role is crucial for effectively managing the escalating computational demands associated with an increasing number of users. Performance evaluations across various integer sizes showed minor delays with non-native datatypes. Nevertheless, the protocol demonstrated efficacy, even in scenarios involving a large number of nodes [98].

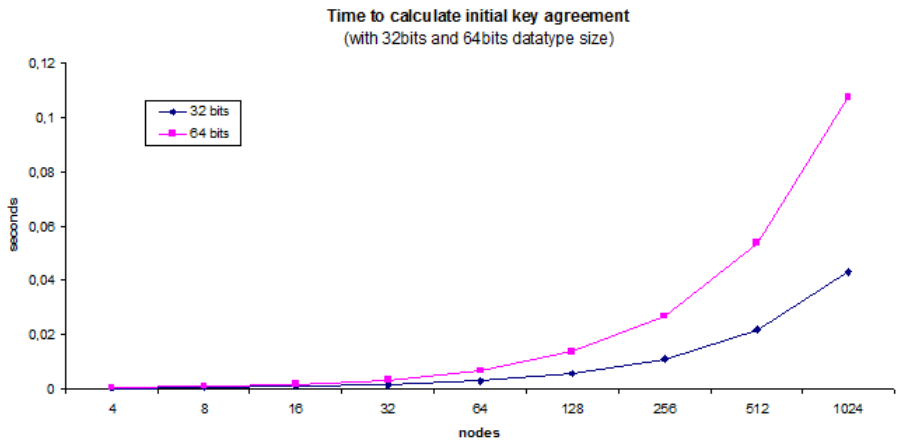

Fig. 4. 7: Time Uc For Calculating The Initial Key Agreement Message [98].

The figure presents the time required by Uc to calculate the initial key agreement message. The figure further highlights the data sizes associated with the key agreement message in the context of 32 and 64-bit data types.

The analysis depicted in Figure 4.8 reveals that the primary bottleneck lies in the memory capacity required to store the message intended for transmission to other nodes. This message is made of a data structure consisting of that includes a list of items, the size of this list is directly related to the to the number of nodes that are part of it.

It is interesting to note that in Figure 4.8 it is shown that even with a considerable number of nodes, specifically 1024, and using keys of size 1024, the size of message remains below 1 MB, specifically measuring at 0.523776 MB. This finding implies that the memory allocation required to store the message is manageable and does not exceed reasonable limits.

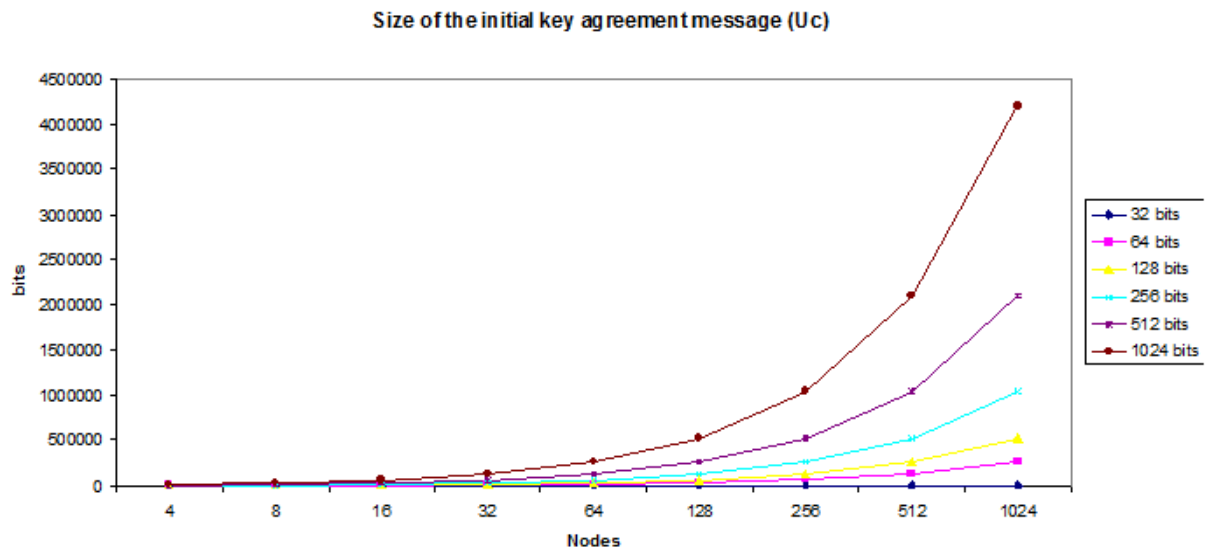

Fig. 4. 8: The Size Of The Message Generated By Uc [98].

The figure presents the sizes of the initial key agreement message created by Uc across various bit lengths. The figure showcases the message sizes for bit lengths of 32, 64, 128, 256, 512, and 1024 bits. This analysis offers a valuable comparison of various message sizes. It provides important insights into the memory or storage needs for transmitting and managing these messages. This information is crucial for comprehending the impact of different bit lengths on system resources and improving the efficiency of the key agreement process.

Segmenting nodes into concurrent subgroups enables faster session key generation by allowing each segment to operate simultaneously. This segmentation offers several benefits, including reduced memory requirements per node, faster group formation, and additional computation for messages received with a session key that need to be re sent with a different session key.

However, it is worth noting that these operations should not negatively affect the overall efficiency of the protocol since the subgroups do not create a bottleneck.

4.2 Testbed and Results for Physical Node Evaluation of the Proposed Protocol

To further evaluate the effectiveness of the proposed protocol, comprehensive tests were conducted in both emulated and physical environments. In the physical node experiments, measures were taken to optimize energy consumption by eliminating unnecessary modules and drivers that were not essential for data collection and transmission to higher-layer nodes. This optimization resulted in a reduced memory footprint and limited the node's activities to essential functions, thereby minimizing extraneous energy consumption and enhancing the node's resilience against uncontrolled attacks. The YOCTO framework [108], an open-source industry-led consortium, was employed to generate dedicated images for the nodes using embedded systems recipes.

The results of the initial key agreement process for physical nodes within a wireless sensor network (WSN) using 64-bit data types are presented in Figure 4.9. The network configurations encompass various sizes, including 1, 10, 100, and 1000 nodes. This analysis primarily focuses on assessing the performance of the proposed protocol.

Since the WSN operates solely within its environment without external communication with cloud servers or other nodes, message latency of less than 1ms is considered negligible. Thus, the depicted results predominantly reflect the protocol's performance within the WSN environment [98].

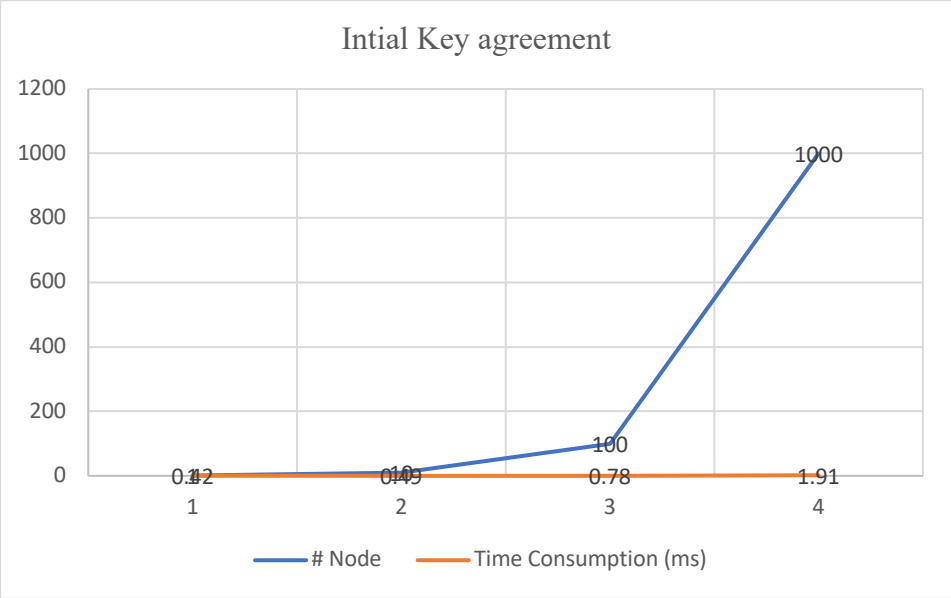

Fig. 4. 9: Initial Key Agreement Results for Physical Nodes in Various Network Sizes With 1024-Bit Keys.

The figure labeled as Figure 4.9 presents the results of the initial key agreement process in a wireless sensor network (WSN) for different network sizes. The findings reveal the durations required for the initial key agreement process in milliseconds. For a single node in the network, the initial key agreement process took 0.42 milliseconds. In the case of a network with 10 nodes, the process duration increased slightly to 0.49 milliseconds. When the network size expanded to 100 nodes, the duration of the initial key agreement process extended to 0.78 milliseconds. Lastly, in a network with 1000 nodes, the process took 1.91 milliseconds to complete. These results offer important insights into the time efficiency of the initial key agreement process in a WSN. They demonstrate the influence of network size on the duration required for establishing initial key agreements and provide valuable information for optimizing performance and scalability in securing communication within similar network environments. Ultimately, the findings contribute to enhancing network efficiency and ensuring secure data transmission within WSNs.

The inclusion of physical node tests played a crucial role in establishing the lower bounds of the proposed protocol's performance and validating its effectiveness under real-world conditions. By conducting experiments on physical nodes, the protocol's behaviour was evaluated in practical scenarios, accounting for hardware constraints and environmental factors. This comprehensive evaluation further solidified the protocol's robustness and suitability for deployment in real WSNs.

The utilization of the YOCTO framework and the generation of dedicated images through embedded systems recipes enhanced the reliability and reproducibility of the physical node experiments. This approach allowed for customized images that optimized performance and ensured compatibility with the specific hardware configurations of the physical nodes. The level of customization and fine-tuning contributed to the accuracy and dependability of the obtained

results. The combined assessment of the proposed protocol in both emulated and physical environments provides a comprehensive evaluation of its effectiveness. The controlled experimentation in the emulated environment and the validation through physical node tests offer valuable insights into the protocol's performance and its potential to address the challenges faced by wireless sensor networks in practical settings.

4.3 Discussion

The impact of the proposed solution is determined by several crucial parameters, including efficient utilization of node energy, network lifetime, ensuring reliable packet transmission, and establishing secure communication among nodes while considering power consumption. It is important to note that this work is specifically developed within the context of agribusiness, where sensors operate either on a single battery or with the assistance of solar power charging cycles. In this unique context, node distribution is heterogeneous and not confined to a perfect isolated area. Factors such as nodes from other farmers, shadow zones in communications due to adverse weather phenomena, recalibration of sensor nodes deployed in hostile conditions, and potential collisions with nodes from neighbouring plots need to be considered. These circumstances emphasize the necessity of dynamically re-electing a head-end (CH) node, not solely based on energy drainage but also in response to these contextual challenges.

To address the challenges mentioned above, a proposed cryptographic method is introduced to prevent data contamination and ensure secure communication while minimizing power consumption. Moreover, an extension to the proposed solution is suggested, incorporating two layers. The first layer involves clustering nodes using a enhanced version of the Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm, where a dynamically elected cluster head (CH) node facilitates efficient data gathering. In the second layer, the sensor nodes are grouped into a lightweight Distributed Ledger Technology (DLT)-based layer, while the CH node maintains the DLT log. This federated approach ensures enhanced packet transmission reliability and network efficiency.

Regarding energy utilization, the evolution of energy drainage is evaluated as the algorithm progresses over time. The proposed clustering approach dynamically elects a cluster head (CH) node using the modified LEACH algorithm in the first layer, while the second layer utilizes a

lightweight DLT-based layer. By distributing the workload between the two layers, optimal energy utilization is achieved, prolonging the network lifetime.

The network lifetime is defined as the point at which the CH node has fully depleted its energy potential for transmitting data packets or even receiving small control packets, without meeting the operative threshold (oTh). It is worth noting that the simulation conducted in our study is based on a two-dimensional space with relatively flat surfaces. However, in real-world scenarios, there are obstructions and uneven surface levels that may affect energy consumption. Therefore, it may be more appropriate to employ a three-dimensional model to accurately capture these environmental factors.

Regarding energy evaluation, we conducted a comprehensive assessment of the energy consumption throughout the algorithm's evolution. The transmission process from node to node follows a fixed pattern, with predefined conditions established prior to the execution. In the field, sensor nodes are meticulously optimized to efficiently collect data and synthesize it into a coherent series. The communication rate remains constant, alongside the fixed packet size. The energy required to transmit a packet is contingent upon the selected routing strategy and the specific packet size.

In light of these considerations, we propose two distinct approaches to address energy optimization. In the first approach, nodes are clustered using a modified version of the Low-Energy Adaptive Clustering Hierarchy (LEACH) algorithm, where a cluster head (CH) is dynamically elected. This clustering mechanism ensures efficient data aggregation within the network. The second approach entails grouping nodes into two layers. In the first layer, sensor nodes function as lightweight nodes within a Distributed Ledger Technology (DLT) framework, while the CH node maintains the DLT log. In the second layer, all CHs are consolidated into a separate DLT-based layer, establishing a second layer of trust. Within this layer, the shared logs and the execution of smart contracts serve to enhance the reliability of packet transmission.

By implementing these approaches, we strive to optimize energy consumption and extend the network's lifetime. It is worth noting that the aforementioned evaluation and proposed strategies are based on a simulated two-dimensional space, assuming relatively flat surface conditions. Nonetheless, in real-world scenarios, the presence of obstructions and uneven terrain necessitates the adoption of a three-dimensional model for accurate energy consumption assessment.

Packet transmission reliability is a critical aspect, as packet loss can occur due to various factors, such as interference from farm machinery operating near the nodes or potential denial-of-service (DoS) attacks. The proposed protocol effectively prevents packet injection, enables rapid discrimination of unwanted packets by CHs, and safeguards against unauthorized access to transmitted information. Additionally, the protocol facilitates the immediate selection of a new node before an imminent node failure. In the proposed extension utilizing DLT, smart contracts can identify attack patterns and disconnect nodes, effectively mitigating the impact of compromised nodes.

Regarding the number of clusters, the decision is influenced by optimizing the locations of cluster heads and considering the trade-off between the number of clusters and energy consumption. Previous research on LEACH for wireless sensor networks in agriculture [105] has revealed that LEACH can lead to an uneven distribution of cluster heads, resulting in concentrated cluster heads in specific areas of the network. This imbalance can cause nodes located far from cluster heads to deplete their energy more rapidly due to the higher power required for successful data transmission.

Based on this understanding, it is recommended to determine the optimal number of clusters to achieve energy efficiency. Research suggests that the ideal number of clusters falls between three and five. When the number of clusters falls below three, cluster sizes become larger, resulting in increased energy consumption for non-cluster head nodes. Conversely, if the number of clusters exceeds six, the smaller cluster sizes lead to more frequent packet transmissions, causing sensor nodes to consume more energy and ultimately reducing the overall network lifetime. Considering these factors, our proposed solution adopts the use of four clusters. This choice aims to strike a balance between cluster size and energy consumption, facilitating efficient data transmission and maximizing the network's lifespan.

The proposed solution aims to address key parameters, such as energy consumption, network longevity, and the reliability of packet transmission, within the agribusiness context. By employing clustering techniques, cryptographic methods, and a layered approach incorporating DLT, the proposal enables efficient and secure data transmission, resulting in extended network lifetimes and reduced energy consumption.

CHAPTER 5
CONCLUSION

The wireless sensor network (WSN) is a fundamental component of the Internet of Things (IoT). It consists of constrained devices in terms of resources, power, processing power, and storage capacity. Sensor nodes may only be able to use extremely short-range radios due to power limitations. Protocols are employed in this instance to let sensor data flow from node to node until it reaches the gateway or sink. Traffic from wireless sensor networks is converted via the gateway into IP protocol traffic for use on conventional data networks. WSN uses light-weight protocols to connect devices and the gateway using dynamic communication.

WSN brought many solutions to applications in different fields, such as remote monitoring systems in the agriculture sector, the industrial sector, the military sector, and the health care sector. WSNs are vulnerable to various security threats due to their nature and constraints. Additionally, sensors are frequently installed in unprotected areas without physical security. We believe that special, enhanced security approaches to securing WSN are necessary due to the variety of WSN applications and potential differences in security requirements.

Numerous algorithms and solutions have been proposed over the past few decades to address security challenges and limitations in WSN, such as group key management. Due to the dynamic change of the topology in terms of group membership joining and leaving, the design of group key management is complex. Furthermore, the scarcity of resources experienced by sensor devices imposes additional constraints that limit their ability to perform computations, store a large amount of data, and manage the high number of sending and receiving keys. Hence, most of the previous methods lacked efficiency while generating and distributing keys while ignoring numerous developments in the realm of communication.

This work presented the state of the art in efficient group key management schemes designed for the WSN using elliptic curve cryptography to enable security services. The proposed solutions reduce the cost of rekeying associated with membership shifting and are highly scalable for group key agreements. The result shows that the protocol reduces the time of operation for the key agreement and the memory footprint, so it can be considered an alternative method to

avoid potential bottlenecks when the group numbers of the network grow rapidly, due to using the method of clustering.

In this study, many routing methods have also been studied and evaluated. Several performance parameters of enhanced LEACH and LEACH are analyzed and simulated for different dynamic parameters. In order to demonstrate the efficacy of the proposed approach in wireless sensor networks, diverse metrics, including the number of rounds, energy consumption, time consumption, number of inactive nodes, and hop counts, are examined for comparison with other routing protocols. The enhanced LEACH routing protocol shows better results compared to other routing protocols.

FUTURE WORK

In the future work of this proposed method, several key areas will be addressed to further advance the research and enhance its practical applicability. These areas of focus aim to overcome limitations, enhance performance, and investigate new possibilities. The following directions will be pursued:

Extensive Experimental Testing in Complex Wireless Sensor Network (WSN) Systems: In order to evaluate the proposed solution's robustness, a series of extensive experimental tests will be conducted within complex WSN systems. Specifically, the performance of the Enhanced Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol will be meticulously assessed against various types of attacks. Additionally, comparative analyses will be performed by benchmarking the modified protocol against alternative routing protocols. It is imperative that these evaluations take place within real-world scenarios to effectively gauge the method's effectiveness in practical deployment settings.

Integration of Machine Learning for Intelligent Decision-Making: An integral aspect of future work involves the seamless integration of machine learning algorithms into the E- LEACH protocol. This integration will facilitate intelligent decision-making capabilities by enabling the analysis of sensor data patterns and network conditions. By employing machine learning models, the optimization of crucial aspects such as cluster formation, energy management, data routing, and resource allocation will be realized. Consequently, this optimization process will lead to improved overall efficiency and enhanced utilization of network resources.

Utilization of Realistic Radio Propagation Models in Protocol Evaluation: However, an analysis of the E-LEACH protocol's performance and simulation studies have revealed the utilization of simplistic and unrealistic models. For instance, the current radio propagation model employed fails to account for obstacles, such as trees within the radio propagation channel, and employs simple radio energy models alongside unlimited transmit power levels for protocol evaluation purposes. To rectify this limitation, it is imperative to incorporate realistic radio propagation models that accurately represent the target environment. The choice of a suitable radio propagation model should be contingent upon the specific application of the WSN. For

instance, in the case of simulating protocols for agricultural applications, vegetation attenuation models can be employed to accurately depict the behaviour of radio waves in such scenarios.

It is believed that by focusing on these three key areas in future research, it is anticipated that significant progress will be made in the proposed method. This progress will help overcome limitations, improve performance, and make the method more applicable in practical settings.

PUBLICATIONS

In the context of this thesis, it is essential to acknowledge that specific content has been previously published in diverse forms, including papers and articles. The purpose of this section is to provide a description of these publications and to clarify the contributions made by the researchers to the field of study.

1. **Distributed Key Management to Secure IoT Wireless Sensor Networks in Smart-Agro**

Author Contributions: Conceptualization, S.M.H., J.A.L.R., and J.A.Á.B.; methodology, J.A.L.R.; software, S.M.H., J.A.L.R., and J.A.Á.B.; validation, J.A.L.R. and J.A.Á.B.; formal analysis, J.A.L.R.; investigation, S.M.H. and J.A.L.R.; resources, S.M.H.; data curation, S.M.H. and J.A.Á.B.; writing—original draft preparation, S.M.H., J.A.L.R., and J.A.Á.B.; writing—review and editing, S.M.H., J.A.L.R., and J.A.Á.B.; visualization, S.M.H.; supervision, J.A.L.R.; project administration, J.A.L.R.; funding acquisition, J.A.L.R. All authors have read and agreed to the published version of the manuscript.

Citation :

Mawlood Hussein, S.; López Ramos, J.A.; Álvarez Bermejo, J.A. Distributed Key Management to Secure IoT Wireless Sensor Networks in Smart-Agro. SENSORS **2020**, 20, 2242. https://doi.org/10.3390/s20082242

2. **A Secure and Efficient Method to Protect Communications and Energy Consumption in IoT Wireless Sensor Networks**

Author Contributions: Conceptualization, S.M.H., J.A.L.R. and A.M.A.; methodology, S.M.H. and J.A.L.R.; software, S.M.H. and A.M.A.; writing—original draft preparation, S.M.H., J.A.L.R. and A.M.A.; writing—review and editing, S.M.H. and J.A.L.R.; All authors have read and agreed to the published version of the manuscript.

Citation:

Hussein, S.M.; López Ramos, J.A.; Ashir, A.M. A Secure and Efficient Method to Protect Communications and Energy Consumption in IoT Wireless Sensor Networks. ELECTRONICS 2022, 11, 2721. https://doi.org/10.3390/electronics11172721

3. Distribución segura de claves en redes de sensores para IoT

Citation:

S. M. Hussein, "Distribuci´on segura de claves en redes de sensores para IoT," in III Jornadas de Doctorado en Informática (JDI'2020), Universidad de Almería 2020. [Online]. Available: http://digital.casalini.it/9788413510835 [Online]. Available: http://digital.casalini.it/9788413510835

REFERENCES

[1] R. Khatoun and S. Zeadally, "Smart cities: concepts, architectures, research opportunities," *Communications of the ACM,* vol. 59, no. 8, pp. 46-57, 2016.

[2] Hampshire. "IOT CONNECTIONS TO REACH 83 BILLION BY 2024, DRIVEN BY MATURING INDUSTRIAL USE CASES." Juniper Research Ltd. https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024 (accessed 2021).

[3] A. Iglesias, D. Santillán, and L. Garrote, "On the barriers to adaption to less water under climate change: policy choices in Mediterranean countries," *Water Resources Management,* vol. 32, pp. 4819-4832, 2018.

[4] Z. Pang, L. Zheng, J. Tian, S. Kao-Walter, E. Dubrova, and Q. Chen, "Design of a terminal solution for integration of in-home health care devices and services towards the Internet-of-Things," *Enterprise Information Systems,* vol. 9, no. 1, pp. 86-116, 2015.

[5] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems,* vol. 82, pp. 395-411, 2018/05/01/ 2018, doi: https://doi.org/10.1016/j.future.2017.11.022.

[6] A. Kumar and G. P. Hancke, "A zigbee-based animal health monitoring system," *IEEE sensors Journal,* vol. 15, no. 1, pp. 610-617, 2014.

[7] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture," *Computers and electronics in agriculture,* vol. 157, pp. 218-231, 2019.

[8] G. Bitella, R. Rossi, R. Bochicchio, M. Perniola, and M. Amato, "A novel low-cost open-hardware platform for monitoring soil water content and multiple soil-air-vegetation parameters," *Sensors,* vol. 14, no. 10, pp. 19639-19659, 2014.

[9] J. Lloret, M. Garcia, S. Sendra, and G. Lloret, "An underwater wireless group-based sensor network for marine fish farms sustainability monitoring," *Telecommunication Systems,* vol. 60, pp. 67-84, 2015.

[10] T. Fukatsu, T. Kiura, and M. Hirafuji, "A web-based sensor network system with distributed data processing approach via web application," *Computer Standards & Interfaces,* vol. 33, no. 6, pp. 565-573, 2011.

[11] Y.-D. Kim, Y.-M. Yang, W.-S. Kang, and D.-K. Kim, "On the design of beacon based wireless sensor network for agricultural emergency monitoring systems," *Computer standards & interfaces,* vol. 36, no. 2, pp. 288-299, 2014.

[12] S. Misra and S. Singh, "Localized policy-based target tracking using wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN),* vol. 8, no. 3, pp. 1-30, 2012.

[13] R. Prodanović *et al.*, "Wireless Sensor Network in Agriculture: Model of Cyber Security," *Sensors,* vol. 20, no. 23, p. 6747, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/6747.

[14] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks,* vol. 54, no. 15, pp. 2787-2805, 2010.

[15] L. Zhang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-Based Authenticated Asymmetric Group Key Agreement Protocol," Berlin, Heidelberg, 2010: Springer Berlin Heidelberg, in Computing and Combinatorics, pp. 510-519.

[16] G. Y. D. Feng, "A Complete Anonymous Group Key Agreement Protocol," in *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, Wuhan, Hubei, China, 24-25 April 2010 2010: IEEE, doi: 10.1109/NSWCTC.2010.217.

[17] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory,* vol. 22, no. 6, pp. 644-654, 1976, doi: 10.1109/TIT.1976.1055638.

[18] S. Sharma and C. R. Krishna, "An efficient distributed group key management using hierarchical approach with elliptic curve cryptography," in *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 2015: IEEE, pp. 687-693.

[19] M. Steiner, G. Tsudik, and M. Waidner, "CLIQUES: A new approach to group key agreement," in *Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No. 98CB36183)*, 1998: IEEE, pp. 380-387.

[20] S. Shin and T. Kwon, "A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes," *Sensors,* vol. 19, no. 9, p. 2012, 2019.

[21] D. Xiao, X. Liao, and S. Deng, "One-way Hash function construction based on the chaotic map with changeable-parameter," *Chaos, Solitons & Fractals,* vol. 24, no. 1, pp. 65-71, 2005.

[22] H.-F. Huang, Y.-F. Chang, and C.-H. Liu, "Enhancement of two-factor user authentication in wireless sensor networks," in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010: IEEE, pp. 27-30.

[23] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "PAuthKey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications," *International Journal of Distributed Sensor Networks,* vol. 10, no. 7, p. 357430, 2014.

[24] R. Amin and G. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks,* vol. 36, pp. 58-80, 2016.

[25] Z. Mahmood, A. Ullah, and H. Ning, "Distributed multiparty key management for efficient authentication in the internet of things," *IEEE Access,* vol. 6, pp. 29460-29473, 2018.

[26] R. Beckwith, D. Teibel, and P. Bowen, "Report from the field: results from an agricultural wireless sensor network," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004: IEEE, pp. 471-478.

[27] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 214-226.

[28] V. Kanakaris, D. L. Ndzi, K. Ovaliadis, and Y. Yang, "A new RREQ message forwarding technique based on Bayesian probability theory," *EURASIP Journal on Wireless Communications and networking,* vol. 2012, no. 1, pp. 1-12, 2012.

[29] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 7-7 Jan. 2000 2000, p. 10 pp. vol.2, doi: 10.1109/HICSS.2000.926982.

[30] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *sensors,* vol. 12, no. 8, pp. 11113-11153, 2012.

[31] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications,* vol. 1, no. 4, pp. 660-670, 2002, doi: 10.1109/TWC.2002.804190.

[32] S. M. Hussein, J. A. López Ramos, and A. M. Ashir, "A Secure and Efficient Method to Protect Communications and Energy Consumption in IoT Wireless Sensor Networks," *Electronics,* vol. 11, no. 17, p. 2721, 2022.

[33] G. S. O. T. Pazynyuk, *Security in Wireless Sensor Networks*. AG Switzerland: Springer International, 2015.

[34] H. B. Mohammad Sadegh Yousefpoor, "Dynamic key management algorithms in wireless sensor networks: A survey," *Computer Communications,* vol. 134, pp. 52-69, 2019, doi: ISSN 0140-3664.

[35] C. Boyd, A. Mathuria, and D. Stebila, *Protocols for authentication and key establishment*. Springer, 2003.

[36] R. Seetha and R. Saravanan, "A Survey on Group Key Management Schemes," *Cybernetics and Information Technologies,* vol. 15, no. 3, pp. 3-25, 2015, doi: 10.1515/cait-2015-0038.

[37] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," *ACM Computing Surveys,* vol. 35, no. 3, pp. 309-329, 2003, doi: 10.1145/937503.937506.

[38] Patrick McDaniel and a. P. H. Atul Prakash, "Antigone: A Flexible Framework for Secure Group
Communication," in *8th USENIX Security Symposium*, Washington,
D.C.USA, 1999, p. 15.

[39] B. Schneier, "Applied Cryptography Second Edition: protocols, algorithms, and source," *Beijing: China MachinePress,* pp. 239-252, 2000.

[40] A. P. Yongdae Kim, and Gene Tsudik, "Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups," in *the 7th ACM Conference in Computer and Communication Security*, Athens, Greece, 2000: ACM, pp. 235–241, doi: 1-58113-203-4/00/0011.

[41] S. Zhu and S. Jajodia, "Scalable Group Key Management for Secure Multicast: A Taxonomy and New Directions," in *Network Security*, S. C. H. Huang, D. MacCallum, and D.-Z. Du Eds. Boston, MA: Springer US, 2010, ch. Chapter 3, pp. 57-75.

[42] H. S. Yacine Challal, "Group Key Management Protocols: A Novel Taxonomy," *International Journal of Computer, Electrical, Automation, Control and Information Engineering,* vol. 2, no. 10, p. 14, 2008.

[43] W. Diffie and M. E. Hellman, "New Directions in Cryptography," in *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, vol. 42: Association for Computing Machinery, 2022, pp. 365–390.

[44] M. A. Simplício, P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed Wireless Sensor Networks," *Computer Networks,* vol. 54, no. 15, pp. 2591-2612, 2010, doi: 10.1016/j.comnet.2010.04.010.

[45] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," Berlin, Heidelberg, 1995: Springer Berlin Heidelberg, in Advances in Cryptology — EUROCRYPT'94, pp. 275-286.

[46] D. G. Steer, L. Strawczynski, W. Diffie, and M. Wiener, "A secure audio teleconference system," presented at the Proceedings on Advances in cryptology, Santa Barbara, California, USA, 1990.

[47] J.-M. Bohli, "A Framework for Robust Group Key Agreement," Berlin, Heidelberg, 2006: Springer Berlin Heidelberg, in Computational Science and Its Applications - ICCSA 2006, pp. 355-364.

[48] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," *Journal of Cryptology,* vol. 20, pp. 85-113, 2007.

[49] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract)," presented at the Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98, Dallas, Texas, USA, 1998.

[50] S. Tingjun, G. Yuanbo, and M. Jianfeng, "A fault-tolerant and secure multi-conference-key agreement protocol," in *2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No. 04EX914)*, 2004, vol. 1: IEEE, pp. 18-21.

[51] P. Porambage, P. Kumar, C. Schmitt, A. Gurtov, and M. Ylianttila, "Certificate-based pairwise key establishment protocol for wireless sensor networks," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, 2013: IEEE, pp. 667-674.

[52] H.-C. Chu, W.-T. Siao, W.-T. Wu, and S.-C. Huang, "Design and implementation an energy-aware routing mechanism for solar wireless sensor networks," in *2011 IEEE International Conference on High Performance Computing and Communications*, 2011: IEEE, pp. 881-886.

[53] I. Ingemarsson, D. Tang, and C. Wong, "A conference key distribution system," *IEEE Transactions on Information Theory,* vol. 28, no. 5, pp. 714-720, 1982, doi: 10.1109/tit.1982.1056542.

[54] K. Becker and U. Wille, "Communication complexity of group key distribution," in *Proceedings of the 5th ACM Conference on Computer and Communications Security*, 1998, pp. 1-6.

[55] K. McKay, L. Bassham, M. Sönmez Turan, and N. Mouha, "Report on lightweight cryptography," National Institute of Standards and Technology, 2016.

[56] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation,* vol. 48, no. 177, pp. 203-209, 1987.

[57] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM,* vol. 21, no. 2, pp. 120-126, 1978.

[58] B. Seok, J. C. S. Sicato, T. Erzhena, C. Xuan, Y. Pan, and J. H. Park, "Secure D2D Communication for 5G IoT Network Based on Lightweight Cryptography," *Applied Sciences,* vol. 10, no. 1, p. 217, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/1/217.

[59] R. Afreen and S. C. Mehrotra, "A review on elliptic curve cryptography for embedded systems," *arXiv preprint arXiv:1107.3631,* 2011.

[60] B. S. R.Devika , T.Sivasubramanian "Survey on Routing Protocol in Wireless Sensor Network," *International Journal of Engineering and Technology (IJET),* vol. 5, no. 1, p. 6, 2013, doi: 10.7763/IJET.

[61] M. Haque, T. Ahmad, and M. Imran, "Review of Hierarchical Routing Protocols for Wireless Sensor Networks," in *Intelligent Communication and Computational Technologies*, Singapore, Y.-C. Hu, S. Tiwari, K. K. Mishra, and M. C. Trivedi, Eds., 2018// 2018: Springer Singapore, pp. 237-246.

[62] D. M. Birajdar and S. S. Solapure, "LEACH: An energy efficient routing protocol using Omnet++ for Wireless Sensor Network," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2017: IEEE, pp. 465-470.

[63] G. Raghunandan and B. Lakshmi, "A comparative analysis of routing techniques for Wireless Sensor Networks," in *2011 national conference on innovations in emerging technology*, 2011: IEEE, pp. 17-22.

[64] C. Nakas, D. Kandris, and G. Visvardis, "Energy efficient routing in wireless sensor networks: A comprehensive survey," *Algorithms,* vol. 13, no. 3, p. 72, 2020.

[65] T. Hu and Y. Fei, "QELAR: A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks," *IEEE Transactions on Mobile Computing,* vol. 9, no. 6, pp. 796-809, 2010, doi: 10.1109/TMC.2010.28.

[66] M. Razzaq and S. Shin, "Fuzzy-logic dijkstra-based energy-efficient algorithm for data transmission in WSNs," *Sensors,* vol. 19, no. 5, p. 1040, 2019.

[67] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on mobile computing,* vol. 3, no. 4, pp. 366-379, 2004.

[68] Z. Zeng-wei, W. Zhao-hui, and L. Huai-zhong, "An event-driven clustering routing algorithm for wireless sensor networks," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, 2004, vol. 2: IEEE, pp. 1802-1806.

[69] N. A. Alwan, "Performance analysis of Dijkstra-based weighted sum minimization routing algorithm for wireless mesh networks," *Modelling and Simulation in Engineering,* vol. 2014, pp. 32-32, 2014.

[70] M. R. Minhas, S. Gopalakrishnan, and V. C. Leung, "Fuzzy algorithms for maximum lifetime routing in wireless sensor networks," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, 2008: IEEE, pp. 1-6.

[71] P. Neamatollahi and M. Naghibzadeh, "Distributed unequal clustering algorithm in large-scale wireless sensor networks using fuzzy logic," *The Journal of Supercomputing,* vol. 74, pp. 2329-2352, 2018.

[72] V. Gupta and R. Pandey, "Modified LEACH-DT algorithm with hierarchical extension for wireless sensor networks," *International Journal of Computer Network and Information Security,* vol. 8, no. 2, p. 32, 2016.

[73] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, and J. Henry, *IoT fundamentals: Networking technologies, protocols, and use cases for the internet of things*. Cisco Press, 2017.

[74] R. Frank, "Understanding Smart Sensors 2nd edn," *Measurement Science and Technology,* vol. 13, no. 9, pp. 1501-1502, 2002.

[75] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. John wiley & sons, 2007.

[76] C. Sabri, L. Kriaa, and S. L. Azzouz, "Comparison of IoT constrained devices operating systems: A survey," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017: IEEE, pp. 369-375.

[77] S. Chen, C. Yang, J. Li, and F. R. Yu, "Full lifecycle infrastructure management system for smart cities: A narrow band IoT-based platform," *IEEE Internet of Things Journal,* vol. 6, no. 5, pp. 8818-8825, 2019.

[78] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations," *Computer Science Review,* vol. 40, p. 100364, 2021.

[79] C. Bormann, M. Ersue, and A. Keranen, "RFC 7228: Terminology for constrained-node networks," ed: RFC Editor, 2014.

[80] O. Vermesan and P. Friess, *Internet of things: converging technologies for smart environments and integrated ecosystems*. River publishers, 2013.

[81] I. Pena-Lopez, "The internet of things," *Itu internet report,* 2005.

[82] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: A top-down survey," *Computer Networks,* vol. 141, pp. 199-221, 2018/08/04/ 2018, doi: https://doi.org/10.1016/j.comnet.2018.03.012.

[83] V. ETSI, "Machine-to-machine communications (M2M): Functional architecture," *Int. Telecommun. Union, Geneva, Switzerland, Tech. Rep. TS,* vol. 102, p. 690, 2011.

[84] C. N. Academy. "IoT Fundamentals: IoT Security." https://www.netacad.com/courses/cybersecurity/iot-security (accessed January 2021).

[85] D. Guinard, M. Fischer, and V. Trifa, "Sharing using social networks in a composable web of things," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010: IEEE, pp. 702-707.

[86] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," in *2010 Internet of Things (IOT)*, 2010: IEEE, pp. 1-8.

[87] P. K. Verma *et al.*, "Machine-to-Machine (M2M) communications: A survey," *Journal of Network and Computer Applications,* vol. 66, pp. 83-105, 2016.

[88] U. Cisco, "Cisco annual internet report (2018–2023) white paper," *Cisco: San Jose, CA, USA,* vol. 10, no. 1, pp. 1-35, 2020.

[89] M. Sneps-Sneppe and D. Namiot, "About M2M standards and their possible extensions," in *2012 2nd Baltic Congress on Future Internet Communications*, 2012: IEEE, pp. 187-193.

[90] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*, 2008: IEEE, pp. 363-369.

[91] H. Chen, "Applications of cyber-physical system: a literature review," *Journal of Industrial Integration and Management,* vol. 2, no. 03, p. 1750012, 2017.

[92] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks,* vol. 38, no. 4, pp. 393-422, 2002.

[93] I. Bekmezci and F. Alagöz, "Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring," *International journal of Distributed Sensor networks,* vol. 5, no. 6, pp. 729-747, 2009.

[94] I. Stojmenovic, *Handbook of wireless networks and mobile computing*. Wiley Online Library, 2002.

[95] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards," *Computer communications,* vol. 30, no. 7, pp. 1655-1695, 2007.

[96] A. A. Khalaf and M. S. Mokadem, "Effects of ZigBee component failure on the WSN performance with different topologies," in *2016 28th International Conference on Microelectronics (ICM)*, 2016: IEEE, pp. 9-12.

[97] A. Kurtoglu, J. Carletta, and K.-S. Lee, "Energy consumption in long-range linear wireless sensor networks using LoRaWan and ZigBee," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017: IEEE, pp. 1163-1167.

[98] S. Mawlood Hussein, J. A. Lopez Ramos, and J. A. Alvarez Bermejo, "Distributed Key Management to Secure IoT Wireless Sensor Networks in Smart-Agro," *Sensors (Basel),* vol. 20, no. 8, p. 2242, Apr 15 2020, doi: 10.3390/s20082242.

[99] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Information and Computation,* vol. 146, no. 1, pp. 1-23, 1998.

[100] H. Harney and C. Muckenhirn, "Group key management protocol (GKMP) architecture," 2070-1721, 1997.

[101] J.-C. Lin, F. Lai, and H.-C. Lee, "Efficient group key management protocol with one-way key derivation," in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*, 2005: IEEE, pp. 336-343.

[102] S. Pote, V. Sule, and B. Lande, "Arithmetic of Koblitz Curve Secp256k1 Used in Bitcoin Cryptocurrency Based on One Variable Polynomial Division," in *2nd International Conference on Advances in Science & Technology (ICAST)*, 2019.

[103] J. Ellul, J. Galea, M. Ganado, S. McCarthy, and G. J. Pace, "Regulating Blockchain, DLT and Smart Contracts: a technology regulator's perspective," *ERA Forum,* vol. 21, no. 2, pp. 209-220, 2020/10/01 2020, doi: 10.1007/s12027-020-00617-7.

[104] P. Saini and A. K. Sharma, "E-DEEC-enhanced distributed energy efficient clustering scheme for heterogeneous WSN," in *2010 First international conference on parallel, distributed and grid computing (PDGC 2010)*, 2010: IEEE, pp. 205-210.

[105] L. M. Kamarudin, R. B. Ahmad, D. L. Ndzi, A. Zakaria, K. Kamarudin, and M. E. E. S. Ahmed, "Simulation and analysis of leach for wireless sensor networks in agriculture," *International Journal of Sensor Networks,* vol. 21, no. 1, pp. 16-26, 2016.

[106] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems,* vol. 11, no. 8, pp. 769-780, 2000, doi: 10.1109/71.877936.

[107] N. Binkert *et al.*, "The gem5 simulator," *SIGARCH Comput. Archit. News,* vol. 39, no. 2, pp. 1–7, 2011, doi: 10.1145/2024716.2024718.

[108] L. Foundation. "YOCTO Project Open Source Embedded Linux Build System, Package Metadata and SDK Generator." https://www.yoctoproject.org/ (accessed February 3, 2017).