

**STORAGE EFFICIENT DYNAMIC KEY  
MANAGEMENT FRAMEWORK FOR WIRELESS  
SENSOR NETWORKS**

Thesis Submitted For the Award of the Degree of

**DOCTOR OF PHILOSOPHY**

**in**

**Computer Science and Engineering**

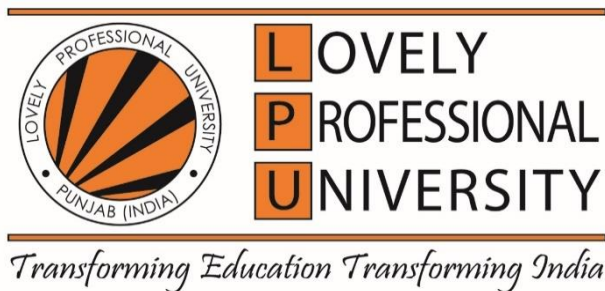
**By**

**Vipin Kumar**

**41600104**

**Supervised By**

**Dr. Navneet Malik**



**LOVELY PROFESSIONAL UNIVERSITY**

**PUNJAB**

**2022**

---

## DECLARATION

I declare that the thesis entitled "Storage Efficient Dynamic Key Management Framework for Wireless Sensor Networks" has been prepared by me under the guidance of Dr. Navneet Malik, Associate Professor, School of Computer Science and Engineering, Lovely Professional University, India. This thesis was written entirely by myself, and no part of it has ever been copied or used as the basis for awarding a degree or fellowship in any other subject.



Date 15 April 2022

Vipin Kumar

School of Computer Science and Engineering

Lovely Professional University

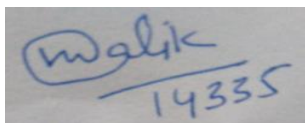
Jalandhar, Delhi G.T.Road (NH-1)

Phagwara, Punjab-144411, India

---

## CERTIFICATE

This is to certify that the thesis entitled "Storage Efficient Dynamic Key Management Framework for Wireless Sensor Networks", which is being submitted by Vipin Kumar for the award of the degree of Doctor of Philosophy in Computer Science and Engineering from the Lovely Faculty of Technology and Sciences, Lovely Professional University, Punjab, India, is entirely based on the work carried out by him under my supervision and guidance. The work reported embodies the original work of the candidate and has not been submitted to any other university or institution for the award of any degree or diploma, according to the best of my knowledge.



Date 15 April 2022

Dr. Navneet Malik

Associate Professor

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab-144411, India

---

## ABSTRACT

Wireless sensor networks are networks of geographically distributed and specialized sensors that monitor and record the physical characteristics of the environment before transmitting the gathered data to a central point. It is a network of tiny, low-cost sensor nodes that communicate wirelessly. These self-contained sensor nodes are built using microcontrollers, radios, batteries, one or more sensors, and interconnecting electronics. Consequently, sensor nodes have limited processing capacity and energy supply, and they are often controlled with energy efficiency rather than performance in mind, resulting in decreased performance. Despite their self-organizing due to radio communications, sensor nodes are designed to be placed and operated unattended. They are not meant to be plugged into a pre-configured architecture. When data from their sensors become available, radio communications allow it to be disseminated throughout the whole sensor network. This offers unrivalled sensing scale and resolution for monitoring-type applications, such as monitoring of natural ecosystems that would otherwise be impossible to achieve. Wireless sensor networks (WSNs) interact seamlessly with the environment in a wide range of applications, providing a transparent interface between human activities and the surrounding environment. WSN, like any other technology, has its advantages and disadvantages. When data is exchanged in an unattended network, it is conceivable that it will be leaked in a way that is detrimental to the interests of the parties involved. There should be a mechanism to check that the network's service is available, as well as a method to differentiate between legitimate and malicious requests in order to avoid the performance of activities that may be potentially disruptive to the network. It is at this point that safety must be considered. The emphasis of this thesis is on information security, specifically the protection against denial-of-service (DoS) assaults. Across all of these industries, energy efficiency is a common thread that runs through them all. At the same time, tamper-proofing the hardware and increasing the processing capability of the sensor nodes may help to enhance security. As a consequence, true WSN security is a delicate

balancing act that is always striving to provide the highest level of protection possible while working with restricted resources.

Wireless sensor networks are essentially wireless networks composed of many tiny battery-operated sensors. These sensors are deployed in harsh environments to collect different types of data, such as temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects or substances, mechanical stress levels on attached objects, and other properties. Recently there have been exploratory growth in the research of wireless sensor networks due to wide applications like health monitoring, environment monitoring, and urban traffic management. Sensor network applications have been used in habitat monitoring, border monitoring, health care, and military surveillance. In some applications security of these networks is essential and needs robust support. For a network, it is very important that nodes in the network trust each other, and the malicious node should be discarded. Cryptography techniques are normally used to secure the networks. Key plays a very important role in network security. Other aspects of security, such as integrity, authentication, and confidentiality, also depend on keys. In Wireless Sensor Network, it is very difficult to manage the keys as this includes distribution of keys, generation of new session keys as per requirements, and renewal or revoking of the keys in case of attacks. In this research, we proposed a Scalable and Storage Efficient Key Management Scheme for wireless sensor networks that establish the different types of keys for the sensor networks. A network key that is shared by all the nodes in the network, a cluster key shared for a group of nodes, and a pair-wise key for every pair of nodes. We analysed the resiliency of the scheme (that is, the probability of key compromise against the node capture) and compared it with other existing schemes. The proposed scheme gives good results as compared to existing schemes in terms of energy, storage, and computation requirements. Proposed is a dynamic key management system that also supports the inclusion of the new node and refreshes the keys as per requirements. The main objective of this research is to improve the security of randomly deployed large-scale wireless sensor networks. This study aims to develop a network that is resistant to node capture attempts and has a strong connection. This approach generates the pairing key for random networks, which can be utilized to build clusters securely.

---

## ACKNOWLEDGEMENT

First and foremost, I am thankful to the almighty for making things possible at the right time. I owe my success to my supervisor and would like to sincerely thank Dr. Navneet Malik for his guidance. I greatly appreciate his support, positive attitude, and his vast knowledge of a wide range of topics. His guidance, not only in terms of giving ideas and solving research problems but also in terms of giving me the freedom to do research in my own ways, has proved to be useful and invaluable. I am deeply influenced by my supervisor's way of guidance and sincerely thankful for standing by my side in tough times.

I very much thank full to my seniors, Dr. Gaurav Dhiman and Dr. Tarun Kumar Lohani Sir, for providing me necessary facilities and resources during the period of my research.

I am also grateful to my friends, colleague, and fellow researchers, particularly Dr. Sukhkirandeep Kaur, Mr. Makul Mahajan, and Dr. Goutam Majumder, for their constructive criticism and suggestions.

I would like to show my gratitude to the entire family of Lovely Professional University, School of Computer Science and Engineering, for providing me with a suitable research atmosphere to carry out my work in the proper time. I am very much grateful to my loving mother, father, and all my family members for the moral support and care that they exhibited towards me during the period of this work.

Finally, I thank God for sailing me through all the rough and tough times during this research work.

---

# TABLE OF CONTENTS

<b>DECLARATION</b>	<b>i</b>
<b>CERTIFICATE</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Wireless Sensor Network	1
1.2 Types of Wireless Sensor Networks	2
1.2.1 Homogeneous versus Heterogeneous WSN	2
1.2.2 Static versus Mobile WSN	3
1.2.3 Flat versus Hierarchical Sensor Network	3
1.3 Wireless Sensor Network Applications	4
1.4 Attacks on Wireless Sensor Network	8
1.5 Sensor Networks Security	12
1.6 Security Requirements	14
1.7 Architectures of Sensor Networks	17
1.8 Key Management	19
1.9 Sensor Network Limitations	20

1.10	Motivation of Research	22
1.11	Thesis Outline	23
<b>2</b>	<b>Review of Literature</b>	<b>24</b>
2.1	Background	24
2.2	Probabilistic Key Scheme	28
2.3	Matrix Based Scheme	31
2.4	Location Based Scheme	32
2.5	Polynomial Based Scheme	34
2.6	Hash Chain Based Scheme	35
<b>3</b>	<b>Problem Identification and objectives</b>	<b>43</b>
3.1	State of Art in Key Management	43
3.2	Research Gap	45
3.3	Objectives of Research	46
3.4	Proposed Methodology	46
<b>4</b>	<b>Scalable and Storage Efficient Key Scheme</b>	<b>49</b>
4.1	Introduction	49
4.2	Network Model	49
4.3	Proposed Scheme	50
4.4	Key Chain	50
4.5	Key Pre-distribution and Key Initialization	52
4.6	Key Setup	52
4.7	Key Renewal	53
4.8	Node Addition	54
4.9	Key Refresh	55



<b>5</b>	<b>Simulation Result</b>	<b>57</b>
5.1	Performance Parameters	57
5.2	Performance Evaluation	61
5.3	Safety Analysis	63
5.3.1	Node replication attack	63
5.3.2	Replay Attack	63
5.3.3	Authentication	64
5.4	Connectivity	64
5.5	Storage Overhead	65
5.6	Resiliency against Node Capture	66
5.7	Energy Consumption	67
5.8	Summary	68
<b>6</b>	<b>Connectivity and Resiliency Analysis</b>	<b>69</b>
6.1	Connectivity and Resiliency	69
6.2	Random Graph Model	69
6.3	Mathematics use in Network security	71
6.4	Resiliency and Connectivity in Random Key distribution	72
6.5	Analysis of Connectivity and Resiliency	74
6.6	Summary	78
<b>7</b>	<b>Key Management for Hierarchal network</b>	<b>80</b>
7.1	Introduction	80
7.2	Network Model	80
7.3	Proposed Scheme	81

7.4 Observation and Analysis	83
7.5 Security Analysis	83
7.6 Simulation Results	84
7.7 Summary	85
<b>8 Dynamic key management for modern Applications</b>	<b>86</b>
8.1 Introduction	86
8.2 Heterogeneous and IoT Networks	86
8.3 Heterogeneous and IoT Networks Applications	87
8.4 Security requirements of IoT Networks	89
8.5 Security challenges	90
8.6 Related Work and Existing Technique	92
8.7 Proposed Scheme	93
8.8 Security framework for Heterogeneous Networks	95
8.9 Security Analysis	97
8.10 Summary	97
<b>9 Conclusions and Future Research Scope</b>	<b>99</b>
9.1 Overall Summary	99
9.2 Conclusion	100
9.3 Future Research Scope	101
<b>BIBLIOGRAPHY</b>	<b>104</b>
<b>LIST OF PUBLICATIONS</b>	<b>114</b>

---

## LIST OF TABLES

1.1	Sensor Network Layer Attacks and Defence	12
2.1	Comparison on the basis of Different Parameters	41
4.1	Notations	50
5.1	Key Parameters	63
6.1	EBS Matrix Example	72
6.2	Key Distribution For Overlapping area	74
8.1	Security Requirements for IoT	91

---

## LIST OF FIGURES

<b>1.1</b>	Berkeley Mote	2
<b>1.2</b>	Distributed and Hierarchical Network	3
<b>1.3</b>	Sensor Network Application	4
<b>1.4</b>	Sensor Network	8
<b>1.5</b>	Taxonomy for WSN security structure	17
<b>2.1</b>	Key Management	27
<b>2.2</b>	Matrix Key Calculation	31
<b>3.1</b>	Flowchart of proposed methodology	47
<b>4.1</b>	Key Chain Pool	51
<b>5.1</b>	Basic NS3 Data Flow Model	61
<b>5.2</b>	NS3 Modules	62
<b>5.3</b>	Connectivity Graph	65
<b>5.4</b>	Resilience to Node Compromise Attack	67
<b>5.5</b>	Energy Consumption	68
<b>6.1</b>	Node Deployment	75
<b>6.2</b>	Connectivity of Random Graph Model	76
<b>6.3</b>	Resiliency of Proposed Model	77
<b>7.1</b>	Network Model	81
<b>7.2</b>	Session Key Generation	82
<b>7.3</b>	Energy Consumption in Key Setup	83
<b>8.1</b>	Smart Building	92

---

## LIST OF ABBREVIATIONS

ADC	Analog to Digital Convertor
BS	Base Station
CDMA	Carrier Sense Multiple Access
CH	Cluster Head
EBS	Exclusion Basis Systems
HWSN	Heterogeneous Wireless Sensor Network
ID	Identity
IoT	Internet of Things
KDC	Key Distribution Centre
KMS	Key Management Scheme
LAN	Local Area Network
MANET	Mobile Ad hoc Network
MAC	Message Authentication Code
MAN	Metropolitan Area Network
NIC	Network Interface Card
QoS	Quality of Service
UID	Unique Identity
VANET	Vehicular Ad hoc Network
WAN	Wide Area Network
WBAN	Wireless Body Area Network
WiFi	Wireless Fidelity
WSN	Wireless Sensor Network

# Chapter 1

## Introduction

### 1.1 Wireless Sensor Network

A sensor is a tiny device with limited processing power, computation power, and memory. A Wireless Sensor Network is a network of sensors that gathers various types of data like temperature, pressure, humidity from the environment and stores it for processing. Examples of such types of data include temperature, pressure, humidity, level, movement, and so on [1]. This information is made available to the sink through the use of a gateway [2]. Even though sensor nodes are deployed randomly, it is important to deploy them with care [3]. Although expanding coverage with a limited number of nodes is feasible, improving network efficiency with many nodes is difficult due to increased collision and interfering signal levels. Sensor network nodes are typically a wireless transponder with an externally or internally mounted antenna, a central processing unit (CPU), a battery, or an integrated energy harvesting device, among other things. Sensor nodes with limited size and cost are also constrained by limitations in energy, memory, processing speed, and transmission capabilities, among other things. To transfer data between nodes in a linked network, routing or flooding may be used [4]. The factors listed below make it very difficult to offer security to these networks. First and foremost, sensor nodes are designed highly resource-limited to economic reasons, rendering the use of public-key ciphers impossible. As a second advantage, data packets are broadcast over the air, making it venerable, and an adversary may intercept the communication data or channel [5].

Sensor devices may vary in size from a shoebox to a dust particle, functioning "motes" with minimal part work successfully. Sensors limited in cost and size are also limited in other resources like energy, memory, processing speed, and communication bandwidth. A wireless sensor node, as shown in figure 1.1, has sensing, processing, a transmitter, and power. Sensors generate analog signals that can be converted into digital signals, and in general, this unit may be connected to another unit through a small storage unit, which will handle the activities required for the sensor node to interact with the other nodes.

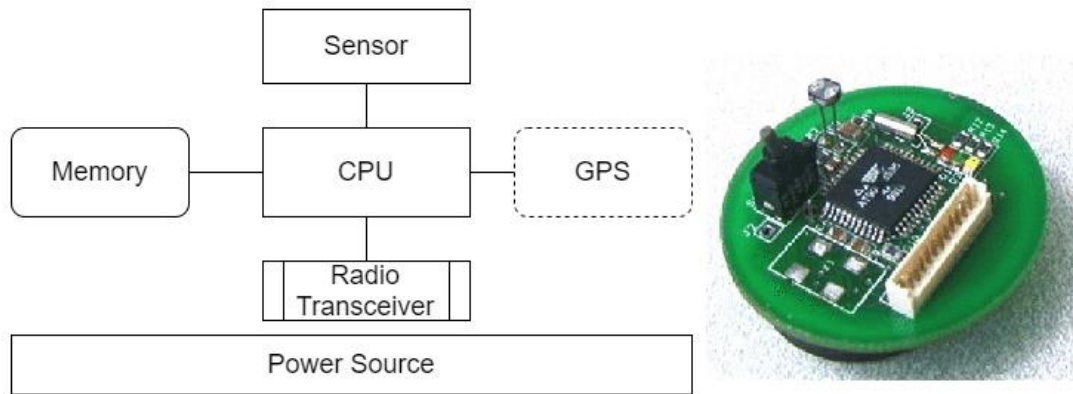


Fig. 1.1 Berkeley Mote

Depending on the application, a positioning system, power source, and conversion tool may be added. Above is a functional block diagram of wireless sensor nodes. These modules offer a highly flexible basis for dealing with the requirements of a wide variety of applications. Depending on how the sensors are arranged, the signal conditioning block, for example, may need to be changed. As a result, a wide range of sensors may be utilized with the wireless sensing node [6]. Additionally, the wireless connection is replaced with an application of one's choice.

## 1.2 Types of Wireless Sensor Network

Wireless networks are categorized on the basis of different parameters that are types of sensors, mobility of sensors, and organization of sensors in the network. In our research, we consider the following types of networks.

### 1.2.1 Homogeneous versus Heterogeneous WSN

Inhomogeneous networks have the same battery energy and hardware complexity as heterogeneous networks, while heterogeneous networks have different characteristics. The long-distance broadcasts to the distant base station, as well as the additional processing required for data aggregation and protocol coordination, will ultimately overwhelm the cluster head nodes in a homogeneous network with static clustering (cluster heads are elected once and remain in place for the duration of the network).

On the other hand, when using a heterogeneous sensor network, you may employ two or more distinct kinds of nodes, each with its own battery life and capabilities. The overall network hardware cost may be reduced significantly by combining more advanced hardware and additional battery energy with a few cluster head nodes. Role rotation is no longer allowed since the cluster head node has been repaired.

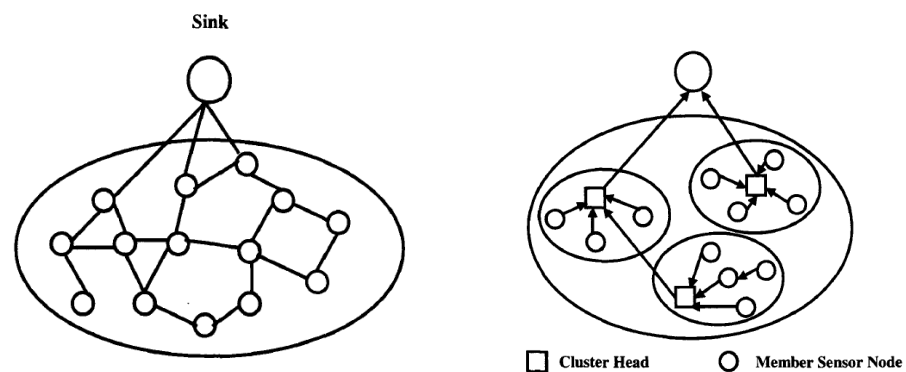
Heterogeneous networks are made up of various sensor nodes, each with its own set of capabilities [7]. Homogeneous networks, on the other hand, provide excellent outcomes in a wide range of applications. Nonetheless, integrating a variety of sensor nodes has been proven to enhance network security and lifespan substantially [8].

### 1.2.2 Static versus Mobile WSN

In static WSN, all the sensors of the network are static and remain at one place all the lifetime. These sensors do not leave their position and collect data from a specific region in their lifetime. In static networks topology of the network is not changed, and generally, routing information is not updated frequently. The mobile node may move from one place to another. Because they may be deployed in any scenario and can deal with abrupt topology changes, mobile WSNs are much more stable than static sensor networks. Both of their applications, however, have been rejected. Many of their applications, such as environmental control and surveillance, are, however, somewhat similar [9].

### 1.2.3 Flat versus Hierarchical Sensor Network

In the flat architecture WSN sink individually collect data from every node, which consumes high energy. Every node sends the data in a multi-hop fashion to the sink. Clustered based network can be used to increase the efficiency of the network and increase life. A cluster leader is chosen, who collects data from all group members and sends it to a sink. If the clustered head has the same capability as a member node, then it is a homogeneous clustered network, as shown in figure 1.2. In some networks, the cluster head may have a high computation, transmission, and storage capacity compared to the member node, which is called a heterogeneous clustered network [10].



**Fig 1.2. Distributed and Hierarchical Network**



### 1.3 Wireless Sensor Network Applications

WSN may be found in several indoor and outdoor locations. While sending data over a network, it is essential to maintain security. Because it's impossible to keep an eye on the sensor nodes/network all of the time, security is the most challenging task in a WSN. It must, however, be secured to prevent an intruder from tampering with data transmission. Security is a significant issue that has to be handled carefully. Wireless networks are utilized in various applications, including health monitoring, environmental monitoring, military applications, and traffic control in cities. In figure 1.3, a soldier is using a handheld device to access the data from the sensors network. Environment surveillance, surveillance systems, immigration monitoring, and universal healthcare are just a few of the applications for wireless sensor networks [11].



Fig 1.3. Sensor Network Application

Network security is essential in certain applications and must be done correctly. To operate properly, nodes in a network must trust one another, and malicious nodes must be eliminated. Cryptography is used to safeguard the network, and the key is essential for various cryptographic techniques. These techniques also rely on the secure and confidential selection of keys. Because WSN includes key distribution, handling keys is difficult. One of the most frequent problems when it comes to establishing security in WSN is key distribution. If every node has the same key and one is hacked or seized by an attacker, the whole network is at risk, yet, if each node has its own key, maintaining all keys will be challenging due to many nodes [12]. An important element

of data, information, and sensor security is key management. Node mobility was not a top concern in many older protocols. For homogenous networks with a fixed Access Point and central node, as well as additional nodes that move around these heads, this paper suggested a Hash Chain-based key management system. This solution stands out when compared to other basic key management systems [13]. The proposed approach improves node capture resistance while simultaneously considering memory limitations.

Industrial automation, patient tracking, medical research, and environmental monitoring are just a few applications that use WSN networks. Battlefield surveillance, environment monitoring, healthcare, and traffic management are just a few military and civilian applications for wireless sensor networks. It's built to handle various high-level data processing tasks, such as detection, tracking, and classification, in a single networked environment [14]. These activities include naming a few well-defined performance indicators such as identifying false alarms or misses, rectifying classification errors, and monitoring quality. Sensing networks are used in many different applications, each with its own set of criteria deployment methods. WSNs and their variants, such as IoT, are often placed to sense data from the environment to take necessary actions to minimize the impacts of the environment on the environment. As a result, they've been utilized in several applications, including military applications, medical applications, environmental applications, structural health monitoring, and a variety of others. Following are the major applications area of WSN

### **1.3.1 Military Applications**

One of the most frequent uses for WSNs is area monitoring. Sensor nodes are deliberately placed over a specified area in this application to monitor many types of events. The installation of sensor networks in border regions to monitor enemy activities so that necessary measures may be taken in the event of an enemy invasion is one example. Military services and applications are scattered over thousands of acres of land. Firearms, ammo, tools, and sensitive information are all examples of "weapons." A large number of soldiers are stationed at national borders and hostile regions to protect civilians and provide them with greater security; however, in the event of an emergency, some mechanism based on modern electronics and technology is required to alert military officers stationed in surveillance areas so that they can

quickly lock down the security and tighten the screws. WSN transmits data wirelessly from one location to another location. In applications like military surveillance, seismic activity monitoring, earthquake detection, and disaster assistance, it has the potential to play a significant role in providing a valuable service in distant areas.

### **1.3.2 Medical Applications**

When used in medical applications, wireless sensor networks (WSNs) allow for remote monitoring of patients' health. In these applications, patient-worn sensors or sensors implanted within their bodies are utilized. Consequently, it may be feasible to build a network of sensor nodes and actors to monitor people's health, particularly the elderly [15]. The gathered data may subsequently be transmitted to medical experts, who will take the necessary measures. Since then, there has been significant progress in developing BAN (Body Area Networks), a subtype of WSN. The increasing popularity of these networks has facilitated the rapid development of telemedicine systems that enable remote monitoring of patients and critical data [16].

### **1.3.3 Environmental Applications**

Among the environmental applications of WSNs are fire detection in forests, agricultural monitoring in fields, and habitat monitoring in natural habitats, to mention a few examples [17]. Forest fire detection systems detect flames mainly by monitoring the temperature of the forest and the presence of gases produced by the fire itself. Sensors are strategically positioned throughout the forest to detect fires. The sensors placed in the agricultural field for agricultural monitoring detect mainly the water and fertilizer requirements of the crops, which are subsequently utilized to manage the crop harvesting process. As an additional tool for monitoring bird habitats, weather sensor networks (WSNs) can detect temperature, pressure, and humidity changes [18].

### **1.3.4 Health Monitoring**

Recent years have seen a significant increase in wireless sensor networks in healthcare systems. The World Wide Web may be used to create a simple but effective system for continuously monitoring the condition of patients. Patients can be watched and monitored in their homes, hospital rooms, and intensive care units in routine or emergencies (ICUs). Even while existing techniques allow for continuous monitoring of a patient's vital signs, they need wired sensors linked to bedside monitors or personal computers and tethered to the patient's bed to be effective. The mesh of wires

surrounding the patient's bed creates a physical barrier that prevents staff from monitoring the patient and causes physical discomfort. Because of the use of WSN, patients may not only move about but can also be watched from a distance.

### **1.3.5 Disaster Detection and Relief**

If natural catastrophes such as floods, fires, tremors, earthquakes, and volcanic activity are discovered early enough, many lives may be spared. The health of rivers and floodplain ecosystems may be monitored with the use of wireless sensors. WSNs have the potential to gather real-time information on wildfires, which may be used to forecast fire behavior and detect flames, among other things. WSNs installed in buildings may detect early indications of an earthquake and other disasters. Tsunamis may be identified using a variety of sensors, including seismic, hydro acoustic, and infrared. WSNs may gather seismic and infrared data in order to keep track of volcanic activity. [19].

### **1.3.6 Industry**

Various activities in the industrial sector are carried out via wireless sensor networks, including property monitoring, asset tracking, and inventory management, among others. Examples of asset monitoring include the implementation of wireless sensor networks (WSNs) at the company's border to identify potentially dangerous storage conditions for its petrochemical goods, as well as the continuous vibration monitoring of its oil tankers engines. To monitor the health of its semiconductor manufacturing equipment, Intel is experimenting with wireless sensor networks (WSNs). Pressure belts, which are sensor networks used by Boeing to monitor pressure distribution on aircraft wing surfaces, are one kind of sensor network. For instance, one situation in which the usage of WSNs becomes critical is when oil pipelines located near the Arctic Circle must be monitored for temperature changes because otherwise, the pipes may explode if not adequately heated [20]. One example of asset monitoring is wireless sensor networks (WSNs) to monitor its railway fleet, which includes both the freight of its railcars in the United States and the locomotives themselves. A number of businesses are now offering solutions for WSN inventory management to their customers.

### **1.3.7 Intelligent buildings**

According to research conducted by the University of California at Berkeley, sensor network technology can reduce energy consumption for commercial building space

cooling (the most frequent energy usage in commercial buildings) by 44 percent. In addition, WSNs have substantial benefits since the cost of setting up various sensors is very less costly. Wireless sensor networks are becoming more popular for collecting essential data, which are currently at the forefront of the most urgently needed research projects. Wireless sensor networks (WSNs) can infer the movement pattern of individuals inside a structure, which is a step toward context-sensitive ubiquitous computing (CSUC).

### 1.3.8 Agriculture

In the agriculture field, WSN is used to monitor the storage conditions of sugar cane beets, for example, or to assess the temperature, moisture level, and soil type in vineyards. If each animal has a node connected to it, the herd can be readily monitored, and the movement of the animals may be managed. WSNs may also be used to track the health of cowherds. A combined application of the sensor network is shown in figure 1.4.

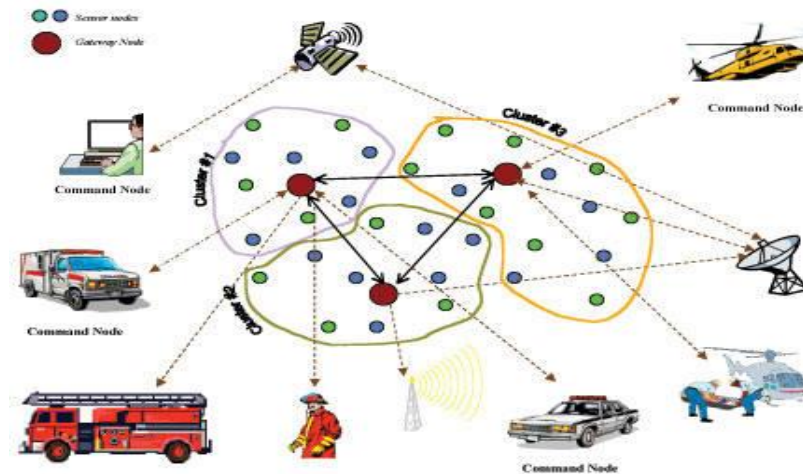


Fig 1.4. Sensor Network

### 1.4 Attacks on Wireless Sensor Networks

Creating a catalog of future threats against WSNs includes understanding the attacks and developing security measures that are appropriate for each categorization of the assaults. For the purposes of this article, the only two types of attacks that can be differentiated are hardware assaults and software attacks, which can be further split into the following types:

- User Compromise
- Hardware Compromise
- Software Compromise

#### **1.4.1 User Compromise**

By deceiving the customers of a WSN into revealing critical usernames and passwords or credentials regarding the machine and these credentials may be compromised. These kinds of attacks are irrelevant until we take into consideration the user behaviors because sensor nodes don't have any or need human operators. [21]. However, even in the case of the base station, many technologies are currently available that may be utilized to ensure the security of the base station and its users. As a consequence, the issue of user compromise does not get any further attention in this study.

#### **1.4.2 Hardware Compromise**

By interfering with a sensor node's hardware, it is possible to get access to the computer software, info, and credentials that are stored inside the node. The attacker may also try installing their program on the malicious node if they have access to it. Despite the fact that sensor nodes are widely recognized to be vulnerable to tampering, these types of assaults are equipment, and compromising a significant portion of a network often involves corrupting a large number of devices at a prohibitively high cost, and because of this, exploiting hardware vulnerabilities to compromise WSNs is no longer a valid attack vector against them on their own.

#### **1.4.3 Software Compromise**

Common vulnerabilities, such as malicious nodes, which affect many other systems, are likely to be present in the software and applications that execute on a sensor node. More importantly, the discipline of developing secure code is well-established, and techniques for assisting in the safe execution of untrusted code are now being developed [22]. Selective forwarding, Gray hole, black hole, and wormhole are all examples of attacks on the network layer of WSNs. Attempting to interrupt network traffic by fabricating, altering, or replaying routing information is one method an attacker may use. In order to attack a routing protocol, the simplest approach is to target the routing information that is already available in the system [9].

The creation of a loop in routing, the attraction or repulsion of nodes to find routes, the production of false error messages, network segmentation, and an increase in end-to-end latency are all examples of network disruptions.

**Selective forwarding:** A WSN is a multi-hop network, and for message transmission to be successful, all of the nodes must be able to correctly pass messages. When a node is compromised, an attacker may force it to selectively forward certain messages while dropping others. [23].

**Sinkhole:** By distorting the routing information of a hacked node, an attacker may make the malicious nodes seem much more appealing to their neighbors, leading to a sinkhole attack. As a consequence, the nodes in the immediate vicinity select the affected node as another node via which their data would've been routed as a result of the intrusion. The fact that all traffic from a significant portion of the network would be routed via the hacked node makes selective forwarding very easy in this kind of attack. [24].

**Sybil Attack:** In the sybil attack, one single hop shows several identities simultaneously in the network. First described as an assault against the aim of redundancy measures in peer-to-peer data storage systems, it has now evolved into various other forms. This assault is described by Newsome et al. from the perspective of a WSN. It is possible to use the Sybil attack to compromise distributed data storage systems and routing techniques. Sybil's attacks algorithm works similarly, no matter what the objective is voting, routing, or aggregate. All of the techniques need the creation of several identities. For the Sybil attack to work, the malicious node would have to assume the identities of many other nodes, resulting in many routes being routed via a single malicious node to compromise the routing protocol [25].

**Wormhole:** The term wormhole refers to a network link with low latency that enables an intruder to exploit the connection to take advantage of two successive network talks. This connection may be established in one of two ways: a single node transmitting messages between two nodes that are near but not neighbors, or a couple of different nodes within the network is a separate region interacting with one another. The second scenario is similar to the black hole assault. A node in one area interacts with another attacking node located in a different network area than the base station [26].

**Black hole and Gray hole:** It is possible to launch a black hole attack by tricking a malicious node into falsely advocating excellent paths to a destination address, either

during the path-finding process or in path change notifications sent to the destination node in proactive routing protocols. Its goal is to either interfere with the finding of the path to packet delivery to the target node in question, depending on the circumstances. Rogue nodes drop data packets regionally and regularly and disturb the proper network routing. So it becomes more difficult to identify this type of attack.

**HELLO flood:** The vast majority of protocols that make use of Hello messages have the erroneous assumption that getting such a packet indicates that the sender is within the radio range of the receiver, which is simply not the case. Many nodes may be fooled into believing they are in close proximity to an attacker using a powerful transmitter. Therefore, the malicious nodes broadcast a fake shortcut to the access point, but all of the nodes that got the HELLO packets try to communicate with the attacker node. These nodes are not vulnerable to assault despite the fact that they are within the transmission range of the adversary.

**Byzantine attack:** In this attack, a hacked node or a group of susceptible nodes can launch a successful assault. Routing loop generation, packet forwarding via inefficient routes, and selective packet deletion are just a few of the attacks that it can do. Byzantine attacks are difficult to detect because networks usually do not exhibit any abnormal behavior while they are being attacked in this manner. Documentation of the information: Unauthorized network nodes may access sensitive or vital information from a compromised network node. This information may contain network topology, node geographic location, or the shortest route between authorized nodes in the network.

**Resource-depletion attack:** A hostile node attempts to drain the resources of other nodes in a network by draining their own resources in this kind of attack [27]. The most often cited resource allocations are battery power, bandwidth, and computing power. Excessive route requests, the creation of beacon packets frequently, or the forwarding of stale packets to other nodes are examples of attacks.

**Acknowledgment spoofing:** The sending of response messages is required by certain wireless sensor network (WSN) routing methods. In order to spoof the acknowledgments, an attacker node must overhear packet broadcasts from its adjacent nodes. Nodes get inaccurate information as a result of this behavior. Considering that certain responses can originate via stations that aren't actually alive, the attacker may



be able to spread misleading information about the state of the nodes across the network [28]. There are numerous more types of attacks that may be conducted against the routing protocols utilized by WSNs in addition to the ones mentioned above. Routing table overflow, poisoning, packet replication, route cache poisoning, hurried assaults, and other similar attacks affect most routing protocols used in WSNs. Several in-depth discussions regarding these attacks have taken place.

Table-1.1. Sensor Network Layer Attacks and Defense

<b>Network</b>	<b>Attacks</b>	<b>Defense</b>
Physical	Jamming Tampering	Spread-spectrum, priority messages, lower duty cycle, region mapping, mode change Tamper-proofing, hiding
Link	Collision Exhaustion Unfairness	Error-correcting code Rate limitation Small frames
Network and routing	Spoofed routing information Selective forwarding Sinkhole Sybil Wormholes  Hello flood attacks Acknowledgment spoofing	Egress filtering, authentication, monitoring Redundancy, probing Authentication, monitoring, redundancy Authentication, probing Authentication, packet leases by using geographic and temporal info Authentication, verify the bidirectional link Authentication
Transport	Flooding Desynchronization	Client puzzles Authentication

### 1.5 Sensor Network Security

There is not a continuous energy source for sensor networks, so it becomes difficult to implement security in these networks. Consequently, another critical problem in WSNs is the development of energy-efficient techniques to extend the network's lifetime. We proposed energy-efficient techniques for enhancing WSN security and network lifetime while lowering energy usage. In survey research on WSNs, security issues are often classified into five to seven categories. Cryptography, safe routing, secure data aggregation, secure data fusion, and location security are the most frequent types. Many

WSN applications need secure connections to function properly. Wireless sensor networks are susceptible to various malicious attacks, such as impersonation, masquerading, interception for deception, and an interception for deception. The growth of unmanaged wireless sensor networks, as well as a lack of physical security, are all contributory causes. As a consequence, the security of sensor networks is essential. Because of the wide variety of applications that WSNs may be employed in, they are anticipated to become widely used in the near future, especially when people are unable to perform continuous monitoring. WSNs may be used for a variety of activities, from basic forest habitat monitoring to very sensitive industrial applications that need high levels of security and dependability. A broad range of wireless sensor network applications needs the transfer of sensitive data. They must rely on safe and dependable data transmission from sensor nodes to the processing center to do their jobs effectively. WSNs also use a wireless channel, which is inherently insecure and vulnerable to a variety of security flaws. These assaults are comparable to those that have occurred on other wireless networks in the past. They may also be used in conjunction with WSNs. Denial-of-service attacks, replay attacks, fabrication attacks, Sybil attacks, hello flood assaults, wormhole attacks, and various variations on the theme are all examples of this kind of attack [29]. Because resource-constrained WSNs are unable to expand due to architectural peculiarities, many well-known security methods developed for conventional wireless networks are ineffective in resource-constrained WSNs. Securing a WSN is an arduous and ongoing effort. In the WSN, certain tasks are roughly mixing all types of issues and assaults.

- Key management is one of the essential jobs in the process of installing and monitoring the safety of the sensor networking system. Secrete keys are used for authentication and confidentiality. Key management systems are important for any system that seeks to maintain privacy, integrity, verification, and other security requirements [30]. Establishing and maintaining keys across legal nodes is the method used, and it enables key update, revocation, and destruction to take place between nodes. Because of the limited resources available, delivering effective key management in WSNs is a difficult task [31].

- The next issue to address is the routing security of a network. The bulk of network layer assaults interrupt message processing and routing, resulting in messages failing to reach their intended destination. External attackers and hacked interns' nodes provide two kinds of threats to routing protocols, both of which are very difficult to identify since the compromised node may generate genuine messages. External threats are the most frequent kind of threat. The bulk of existing routing protocols for WSNs are either security-deficient or do not include any security features of nodes [1].
- The third point to consider is the prevention of denial of service (DoS). Denial of service (DoS) may be defined as any occurrence that reduces or destroys the network's ability to perform the tasks intended. DDoS may be caused by a variety of reasons, including hardware failures, code mistakes, resource depletion, environmental conditions, or any complex interplay between these variables. Disruption of service (DoS) attacks prohibit or decrease the usage of computer resources, disrupt or delay services, cause the network to become unavailable, and isolate legitimate users from a network are all possible outcomes.

### **1.6 Security Requirements**

It is possible to launch passive or active attacks on wireless connections in MANETs and DSNs at the same time, and nodes may wander in unfriendly settings where they are vulnerable to capture and manipulation while connected to the network. Therefore, security is essential in these networks [32]. The membership node necessitates the deployment of flexible and adaptive security solutions to accommodate network topology changes. They must also be scalable in order to keep up with the growing number of nodes on the network. Although cryptographic methods are effective at protecting data as it travels over a network, their effectiveness is contingent on the proper management of key pairs and other cryptographic keys.

It is the way of constructing a secret key among a transmitter and a receiver that is referred to as key management. Given the lack of stable network topology and the precise nature of the security risks that face mobile and wireless communications, conventional key management solutions (such as those relying on Reliable Third Party

candidates) are not very well suited for MANETs or DSNs [33]. This research will look at one of the most significant key management approaches proposed for MANETs and DSNs in recent years, as well as some of their shortcomings. In terms of aims, the WSN's security mechanism goals are similar to those of other networks. They may be summarized in a nutshell as follows:

### **Availability**

When it comes to availability, the preservation of resources even when it's prohibited is what it's all about. Attacks on service availability may result in the loss of or a decrease in the availability of the services targeted. It is possible that the automatic use of countermeasures such as authentication and encryption will help to minimize some of these dangers. Others, on the other hand, require some kind of action to be taken in order to avoid disturbance. In the face of a wide variety of threats, availability guarantees that network services continue to function properly [34]. On the other hand, it may cause problems with the network's routing protocol and the continuation of services at the network layer. A third time, a malicious adversary may knock down high-level services like key management and authentication services at the highest echelons of the organization [35].

### **Confidentiality**

It is important to maintain confidentiality since it guarantees that particular information is only readable or available by those granted access. To put it simply, it safeguards data from passive assaults. It is necessary to maintain confidentiality while transmitting sensitive material, such as military secrets. The dissemination of such knowledge to adversaries may have catastrophic repercussions, as in the case of ENIGMA [36]. Additionally, information about routing and packet forwarding must be kept secret so that the adversary does not have an edge in identifying and finding their targets on a battleground [37].

### **Integrity**

Data or communication integrity guarantees that only authorized parties may alter the data or messages being delivered. It also ensures that a message is never corrupted throughout the transmission process, which is critical. Integrity, like confidentiality, may be applied to a stream of communications, a single message, or certain fields within a message, among other things. Complete stream protection, on the other hand,

is the most practical and straightforward technique for safeguarding rivers. It is possible to guarantee that messages are received precisely as they were sent, with no duplicates, insertions, changes, reordering, or replays, with the assistance of a connection-oriented integrity service, which is responsible for dealing with a high number of messages at the same time. Data deletion and destruction are also included in the scope of services for the integrity service. As a result, it can handle simultaneous communication flow manipulation and service rejections.

### **Authentication**

Technology for authentication guarantees that only authorized parties have access to, and are able to supply data. Its primary aim is to guarantee that communication is genuine and truthful. According to the standard, a single message, such as a warning or alarm signal, is intended to reassure the receiver that the message came from the source that it purports to have originated from. A malicious opponent might mimic a node and get unauthorized access to resources and sensitive information while simultaneously interfering with the functioning of all other nodes if there were no authentication measures in place.

### **Non-Repudiation**

In the absence of non-repudiation, neither the sender nor the receiver may contest message reception. As a consequence, when communication is delivered, the receiver may show that the message was sent by the designated sender. On the other side, after sending a message, the sender may show that the message was received by the designated receiver. Non-repudiation is helpful for locating and isolating infected network nodes. Non-repudiation allows node A to accuse node B of misusing the message and convince other nodes that B has been hacked if it receives an erroneous message from node B.

### **Scalability**

It is not directly related to security, but it is a major issue with far-reaching consequences for security services. Depending on their scale, ad hoc networks may include hundreds or even thousands of nodes. To deal with such a large network of computers, security solutions must be scalable. In this scenario, an attacker may compromise a newly installed network node and exploit it to gain unauthorized access

to the whole system. It is simple to start an island-hopping attack by exploiting a single vulnerability in a distributed network [38]. A taxonomy of different sensor networks attacks is given in figure 1.5.

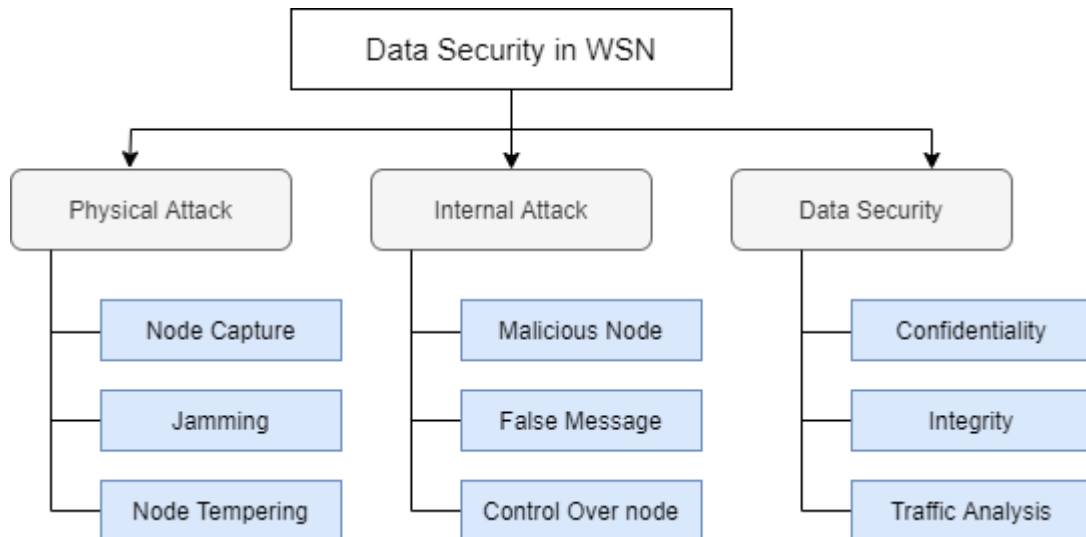


Fig. 1.5: Taxonomy for WSN security structure

### 1.7 Architectures of Sensor Networks

WSANs are being developed in order to incorporate the automated element of WSNs into the whole system. A WSAN is comprised of one or more sensor nodes as well as a restricted number of resource-rich nodes. Certain nodes are charged with taking action in the surrounding environment based on the information gathered by the sensors. Because the main nodes are capable of responding to a detected event in real-time based on the data received from the underlying sensor nodes, it is not necessary to transmit the sensed data to the base station for processing before responding to the detected event. As a result, WSANs enable the adoption of environmental actions in a timely manner. In order to explain the architecture of any network, it is necessary to first specify the following facts: the network's structure, its components, and the delegation of responsibility. To put it another way, the process of developing a new architecture involves answering the following main questions: how will the network nodes be organized, and what kind of nodes will be used to construct the network. Several network designs for wireless sensor networks (WSNs) have been proposed in the literature, whereas mainly two network architectures for wireless sensor networks

(WSANs) have been proposed. WSN makes use of the sensor network architecture, which is a kind of computer network. It is possible to use this kind of architecture in a number of contexts, including hospitals and schools, as well as roads and buildings, and in a variety of applications, including disaster recovery and emergency response management, among others. These architectural designs may be divided into the following categories, which can be used to organize them

### **1.7.1 Flat architecture**

In a flat network architecture, all of the sensors have similar capabilities, implying that all of the sensors are considered peers. When communicating between any sensor node and the base station (BS), a multi-hop path is utilized, with adjacent nodes acting as relays between the two. Responding to the situation, the nodes send data to the base station, which records the information. Each sensor node in a flat network serves the same function as the others and uses roughly the same amount of power from the batteries. In such networks, data aggregation is done via the use of a data-centric routing strategy, which may be summarized as follows: This technique involves the sink sending a query message to the sensors, and then the sensors sending back response messages to the sink if they have data that corresponds to the query. The exact application being evaluated at the moment dictates the choice of a certain communication protocol [39]. The rest of this paragraph explains these methods in detail, highlighting their merits and drawbacks as applicable.

### **1.7.2 Clustered Architecture**

In this kind of architecture, the "Leach Protocol" is utilized, and it is reliant on it since it makes use of clusters. Separate sensor nodes are combined into clusters, which are then reliant on the "Leach Protocol." The term "Low Energy Adaptive Clustering Hierarchy" relates to the protocol "Low Energy Adaptive Clustering Hierarchy" [40]. Some of the most significant aspects of this process are as follows. This is a two-tier hierarchical clustering architecture with a central ring of nodes. This distributed technique is used to arrange sensor nodes into groupings known as clusters in this context. The head nodes of each cluster will generate the TDMA (Time-division multiple access) plans, which will be developed independently of the others. It employs

the Data Fusion concept to decrease the network's energy usage. It is popular due to the data fusion feature of this kind of network architecture. Each node in a cluster may interact with one another through the cluster's head to get data. The data collected by the clusters will be sent to the base station by all of them working together. An independent and autonomous distributed method is used to create a cluster and choose its heads inside each cluster.

### **1.7.3 Layered Architecture**

The network nodes in a layered architecture are grouped into layers, which are then organized into layers. Each layer is typically given a set of tasks. SASNet is a self-healing layered architecture designed for military applications. Sensor Network with Self-Contained Devices Tier-1 nodes in SASNet is made up of many resource-constrained sensor nodes that are in charge of detecting events of interest and transmitting the information to tier-2 nodes, also known as fusion nodes, as soon as they are discovered. Tier-2 nodes handle more sophisticated tasks, including database synchronization, cluster building, application logic formulation, and commanding sensor nodes in response to user requests or queries. Tier 2 is made up of fusion nodes. Finally, tier 3 houses the management of the most powerful nodes. Because of their battery-powered functioning, sensor nodes have no power or processing restrictions. Sensor nodes are spread across various levels based on their distance from the BS, with each layer having many nodes in its own right. An integer  $R$  separates each layer from the preceding layer. The first layer is positioned  $R$  away from the BS, the second layer is positioned  $R$  away from the first layer, and the third layer is positioned  $R$  away from the second layer. If nodes are within range of each other, all higher-level sensor nodes will communicate with lower-level sensor nodes. Because transmission energy is directly proportional to distance  $R$ , and  $R$  is the shortest distance between layers, transmitting data between the various levels of the layers at the shortest feasible distance between them will need the least amount of energy.

### **1.8 Key Management**

Data and information, which do not exist in a physical form, must be protected while sent across computer networks. It is difficult to prevent unwanted individuals from listening in on conversations and impersonating authorized parties via a wireless



network medium. It isn't easy to do so via a wired connection. Several mathematical operations are performed on these streams of bits before they are sent to the recipient via a wired or wireless channel to keep them private from prying eyes. If an opponent is familiar with mathematical processes, they may break private information with relative ease. When an attacker does not have knowledge about particular situations, the adversary can guess them using previously known mathematical operations of cryptography. In the absence of an existing mathematical operation, a pair of communication nodes must agree on new mathematical operations and their inverses each time they interact, increasing the difficulty of the issue even more. Random keys are utilized to make this issue easier to understand. Only authorized parties have the ability to decode or get access to the original data or information. The employment of mathematical functions that are safe and have inverse operations enables us to disclose original information with the assistance of keys. This is particularly useful for cryptography. Keys must be handled safely and effectively if they are to be made accessible for use in any connection at any time. The features, constraints, and applications of a computer network influence the key management of that computer network.

Key management is the main engine for confidentiality and authentication. It is possible to describe key management as a collection of procedures and mechanisms that facilitate the creation of keys and maintain continuing keying relationships between legitimate parties per a security strategy. As a result, one of the most difficult issues in WSN is securing secure connections between nodes. Because of wireless sensor networks' energy, compute, and storage constraints, an asymmetric key cannot be used in WSN [41]. Although specific public key methods are used in WSN, most academics believe that these approaches are still overly burdensome compared to current sensor technology because they impose significant communication and processing overhead.

### **1.9 Sensor Network Limitations**

Public-key methods like Diffie-Hellman [17] and RSA are often considered inappropriate for implementation because of the low computational and power resources available on sensor nodes. It takes a sensor node to perform these activities currently ranges from a few seconds to several minutes [42].

Sensor nodes in open or hazardous areas may be utilized for a variety of reasons (such as public buildings or front combat zones). Furthermore, since so many sensor nodes are being deployed, each one must be cheap, making it hard for producers to produce them adulterate. As a result, hostile physical attacks on sensor nodes are a possibility. In the worst-case scenario, an attacker might capture a node and corrupt the encryption techniques without being detected.

There was a scarcity of information on post-deployment settings prior to deployment. Suppose sensors are distributed at random (for example, by aircraft). In that case, it will be impossible to predict which nodes will be within communication range of each other after the network has been deployed, resulting in an insecure network. Many sensor nodes make forecasting each node's position in advance prohibitively expensive, even if the nodes are manually placed. As a result, no assumptions about which network nodes will be immediate neighbors should be made when designing a security protocol [43].

There is a limit to the amount of RAM accessible. The key-storage memory of a node is limited, and it cannot establish unique keys with every other node in the network. There is a limit to the bandwidth and transmission power. The available bandwidth on a typical sensor network platform is very limited. The transmitter on the UC Berkeley Mica platform, for example, has a bandwidth of 10 Kbps and a packet size of approximately 30 bytes. Data transmissions are often faulty, making big blocks of data transmission especially costly.

Because the system is too reliant on base stations, it is vulnerable to hacking. Base stations, which are both scarce and costly, are required for sensor networks. As a result, it may be tempting to rely on them as a trustworthy source. On the other hand, this encourages base station attacks and restricts the security protocol's deployment to a limited extent [44].

WSN security researchers found that the security paradigm in WSNs should concentrate on tolerance and resilience rather than perfect security. In the face of an attack, a WSN should exhibit elegant performance degradation [45]. If it cannot avoid them completely, it should try to recover as much as possible once they have faded. There is no official consensus on this, although it is widely accepted that in an ideal scenario, the system's performance under assault should degrade at a rate proportional to the ratio

of compromised nodes to total network nodes. The final barrier is provided by people rather than technology. In reality, communication protocol and security experts work within their own separate areas, with limited potential for collaboration [46]. According to an IBM Security Intelligence Services report that examined data from 500,000 electronic devices, many mobile phones, handheld computers, wireless networks, and embedded computers (increasingly being used to run basic automobile functions) are susceptible to virus infection and at risk of infection. As a result, security is still widely viewed as an unresolved issue despite considerable progress in other areas. Many problems remain unresolved or only partially addressed in many instances. On the plus side, this differentiates and fascinates security researchers who work on wireless sensor networks.

### **1.10 Motivation of Research**

Sensors are employed in a variety of applications where nodes are randomly distributed. The nodes in the network communicate via wireless means, and since WSNs are generally installed in unattended areas, the security dangers in the network are greater. In the literature, numbers of protocols and techniques are available to protect the WSN, but the nodes are highly resource-constrained, and providing security for WSN is still a challenging task. The attackers' complexity differs on the basis of the level of sensitivity of the data transmitted over the air. Since the security requirements differ from one application to another, no unique security protocol/ technique that fits all types of deployment/ application is available. It is possible to categorize the degree of security needed for the WSN into three categories, which are as follows:

- Low level or minimum level security
- Standard or medium-level security and
- High-end security

Further classifications may be made depending on the abilities of the attackers who are present in the network. Essentially, the sophistication of the attackers varies according to the sensitivity of the information sent over the air. For example, there may be no attackers present in the network in the agricultural monitoring apps at any one point in time. On the other hand, attackers' capacity to operate in the military and healthcare applications is challenging to assess. However, in applications such as habitat

monitoring, process management software, and other similar ones, the attackers' capabilities are in the range of low to moderate.

Because security needs vary from one application to another, no one security protocol or method can be applied to all kinds of deployments or applications. In a similar vein, key management varies from application to application depending on the security needs since not all apps require the same degree of protection. To choose and design the kind and complexity of key management, it is required to understand the degree of security needed, the availability of hardware configurations, and the network's architecture.

### **1.11 Thesis Outline**

In the rest of this thesis, chapter 2 discusses a literature study and discussion on the necessary background knowledge of relevant key management systems of WSNs. The goals of the thesis and the research objectives and methods are discussed in Chapter 3. The next chapter 4, provides a detailed explanation of the design framework, development environment, and assumptions used. Chapter 4 also contains specifics about our suggested key management system. The findings of the simulation research and the performance analysis are presented in Chapter 5. Chapter 6 discusses connectivity and resilience. In Chapter 7, a scheme implementation for clustered-based networks is provided, and chapter 8 extends the proposed key scheme to IoT. Finally, in Chapter 9, closing comments and future directions are given.

## **Chapter 2**

### **Review of Literature**

#### **2.1 Background**

Since the beginning of computer networks, key management has been a major study topic. Previously, research has mostly focused on computer hardware security. Before creating a computer network, machines should have been emphasized to be supplied with encryption software that protects both the computers and the passive components of the system, and this technology greatly enhanced the security of the system. It wasn't easy to keep track of all the connections since data was transferred between computers and sensors through a network link. Furthermore, the frequency of cyber-attacks on personal information has increased significantly. As a result of these reasons, academics began looking for methods to improve security, particularly key management.

As computer networks increased in size, data was transmitted between previously disconnected devices. The Diffie–Hellman key exchange method was developed to secure such communications [47]. If Alice chooses a random integer  $x$ , Bob will select another random value, such as a different number  $y$ . They then utilize Alice and Bob's hidden values to create a secret key that only they know about. This protocol was rendered useless in enabling two strangers to interact safely due to a lack of an authentication method. Rivest came up with the idea of Public Key Cryptography after inventing the Diffie–Hellman protocol, which quickly became famous. The protocol's name is Rivest, Shamir, and Adleman protocol (RSA). Each computer in the RSA technique is in charge of calculating the public and private keys. If the public key is known, anything is encrypted with it may be decoded with the matching private key. Many algorithms ensure to send data securely but do not offers a method for authenticating the other party using their public-private key pairs. Kerberos was developed in the early 1990s when the internet was exploding with new applications and services [48]. Kerberos authenticates users via the use of symmetric cryptography. In order to ensure that communication keys are authentic, an impartial third party verifies them and distributes them to the parties involved in the communication. One communication party may be assured of the legitimacy of the other communicating party when Kerberos is used. Kerberos ensures that data packets transferred from sender

to recipient are not repeated, and that information is kept private. Apart from these benefits, Kerberos has several disadvantages [49]. As the internet became more popular in the 1990s, several internet-based apps were created. One of the protocols used by these apps was Simple Mail Transfer. All application-level protocols in the internet's architecture rely on transport layer protocols for communication. TLS, created in the late 1990s, creates secure connections between communication parties by using trusted third parties and public-key cryptography principles.

The size and cost of processors, memory, and antennas, all utilized in computing and communication, have been reduced. This is still happening today. Older versions are phased out when new models with better capabilities and smaller sizes are brought to the market. Their value plummets as their pricing soon becomes outdated. Because of the smaller size and lower cost of technology, computer systems can now monitor certain behaviors, phenomena, or biometrics from the human body [50].

The battery life of sensor nodes is also limited. In certain circumstances, such as on battlefields, it is impossible to recharge their batteries. All observed data in wireless sensor networks is sent to a central computer unit known as the base station. During the normal operation of wireless sensor networks, some nodes go out of power and may die, and new nodes join the network.

As a result of the advent of the internet, network security research has increased, resulting in the development of security solutions for computer and sensor networks. Traditional security methods need the use of computer power and/or memory in today's sensor networks. Sensor nodes use a significant amount of energy during the collection, transmission, and reception of data. Basic sensor networks are governed by some concepts, whereas clustered sensor networks are governed by others. The importance of understanding the various kinds of security threads in sensor networks must be understood first before diving into key management methods. It will be easier to comprehend the advantages and disadvantages of different key management systems we first research security risk.

WSN is built on the ability of communication devices, battery power, and sensor nodes to communicate in a wireless environment over a limited region. Due to energy and memory limitations, the development of a fully functional network must be well-organized. There are a number of approaches available in the current literature for such

key management systems. The exchange of private and public keys, as well as key distribution throughout the network, are the most important. A single piece of incorrect data may change the way a network is expected to operate. The integrity of the data must be preserved. Data should not be altered, and accurate data should be sent to the user.

It is impossible to ignore the importance of key management to implement security techniques in networks. The conventional key management techniques mentioned in the background section come in useful in all of these situations. We'll start with the most basic key management choices and work our way up to more complex ones as time allows. We will also provide some of the most pertinent survey findings. Certain characteristics should be compatible with a key management system for a wireless sensor network. The strength of any key management method in wireless sensor networks is determined by the number of such characteristics existing in the system at any one moment. It is essential to preserve the integrity of a secret key in addition to guaranteeing authenticity and secrecy. When we discuss integrity, we mean that the adversary should never create or alter a key. To handle this scenario, the key management system must be scalable. Finally, wireless sensor networks are, by definition, dynamic systems [51]. Older nodes will die when they run out of energy, and new nodes may be added at any time throughout the process. In such cases, a wireless sensor network key management system that is sufficiently flexible should be used.

All key methods for WSN should take care of constraints presented by the sensor nodes themselves, in addition to providing the required level of security. Sensor nodes have no idea where they will be put ahead of time. As a result, they have limited bandwidth, memory, and computing power. The shortcomings of the technology are compounded due to limited energy sources. The public key technique cannot be used in a system with power constraints. Asymmetric key management techniques require a significant amount of energy from the sensor nodes in order to do complex mathematical calculations. Because sensor nodes may only broadcast data over short distances, many sensor network data collecting techniques depend on networking to collect data. This is done to prevent extraneous communication. It's possible that a key management system may fall short of all of the criteria mentioned above.

A network is protected by a single group key. It is the market's most user-friendly key management solution for wireless sensor networks. Instead of utilizing several keys, each sensor node is assigned a single key before deployment. To communicate across all sensor nodes, a single key is utilized. In contrast to previous systems, this method requires very little storage, computation, and transmission power. In WSN pair-wise key is the most secure. Each node on the network is assigned a key that allows it to interact with any other node on the network. This technique guarantees network confidentiality and authenticity while also enabling the removal of an affected node. Another benefit of wireless sensor networks is that nodes do not need to communicate with one another.

In the literature, key management in wireless sensor networks has garnered a lot of attention, and many methods have been suggested. Many early research articles addressing major management problems for heterogeneous sensor networks have been published [18], providing a categorization of symmetric key management systems. All key management systems may be classified into two categories: probabilistic and deterministic, as shown in figure 2.1. Each pair of neighbouring nodes may create a secure direct connection, ensuring that the system covers the whole secure connectivity coverage area.

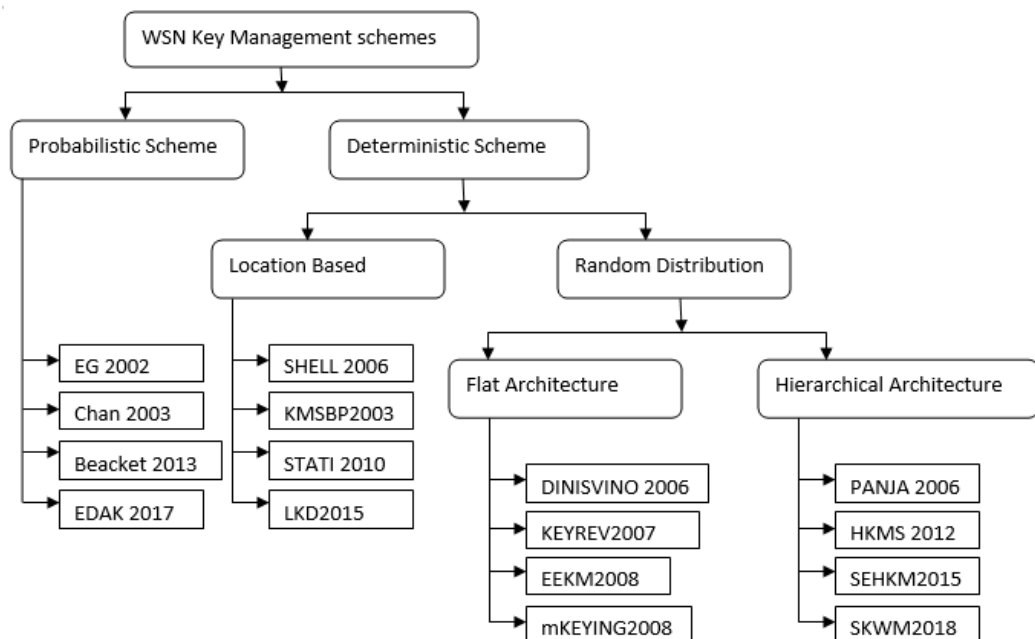


Fig. 2.1 Key Management Schemes



In probabilistic systems, communication between nodes is not guaranteed since it relies on shared keys between nearby nodes. Many key management methods for WSN have been suggested, and they may be classified based on various criteria such as symmetric or asymmetric cryptography, pair-wise or group-wise, centralized or distributed, and dynamic or static. WSN key management systems have been classified in a variety of ways. Some of the classifications can be:

- On the basis of the probability of a common key
  - a) Probabilistic schemes
  - b) Deterministic schemes
- Depending on whether you want to protect pair-wise or group-wise connectivity
  - a) Pair-wise key schemes
  - b) Clustered schemes
- On the basis of memory requirements
  - a) Storage inefficient
  - b) Storage efficient
- Depending on whether key management is the duty of a single node or many nodes
  - a) Distributed schemes
  - b) Centralized schemes
- Based on the cryptography technique
  - a) Asymmetric cryptography
  - b) Symmetric cryptography
- Depending on whether or not node keys are changed throughout the course of the lifespan
  - a) Static schemes
  - b) Dynamic schemes
- Depending on whether or not a location is used as a criterion for key pre-distribution
  - a) Location-dependent
  - b) Location-independent

Various kinds of sensor networks have suggested numerous key management methods in the literature. In the year 2002, Eschenauer and Gilgor [52] proposed the basic key pre-distribution scheme.

## **2.2 Probabilistic Key Scheme**

Random graph probability features must be utilized to get the results. As a result, this method creates a large number of random keys and saves them in the key pool. Following that, each sensor node is given a random key chain made up of key pairs.

They are currently in if they broadcast their keyring and key information to their neighbour's within the wireless communication radius. They may be able to recognize common keys. A new route key must be created for each pair of nodes separated by two or more connections, as well as for every node that does not share a key. This technology's use may be split into three different steps:

**Step 1: Key Initialization.** EG scheme is the term given to this pre-distribution plan since the founders, Eschenauer and Gligor, were the ones who came up with the idea. During the deployment process, keys are kept in the node. A random selection of  $K$  keys is made from a large pool of  $P$  keys and saved in a database shared by the node and base station. Each network node's memory has  $K$  keys, which are selected at random from the set  $P$ . Its keyring comprises a collection of  $k$  keys stored in the node's memory. To be more specific, the number of keys in the key pool with  $P$  keys is set in such a way that any randomly chosen subgroups of number  $\min P$  will share at least one key with a chance of  $p$ , independent of the number of keys in the key pool.

**Step 2: Shared Key Discovery.** Every sensor carries out the process of searching for shared keys with its neighbours. Before deployment, each key is given a brief identifier, and each node broadcasts a list of the IDs associated with it. It is possible for the nodes that identify that they have a shared secret in their key ring to then use a challenge-response protocol to verify that their neighbours really hold the secret. The shared key is then used as the link's key, and the process is repeated.

**Step 3: Path Key Establishment.** When nodes cannot locate shared keys in their key rings, they may opt to establish a route key with nodes in their immediate proximity. If the network is linked, a route from a source node to a nearby node may be found, and vice versa. The source node then creates a route key and transmits it encrypted to the destination node through the previously established path. Before each sensor node is installed, a set of  $m$  keys is given to it, and these keys are taken from a key pool  $S$ . This collection of  $m$  keys is stored on the node's keyring. It is feasible to accomplish this by ensuring that randomly generated sets of size  $m$  in  $S$  each give at least a single key with a possibility of  $p$  and choosing a key pool with a sufficient number of keys. After the sensor nodes have been installed, a crucial configuration step must be performed. The nodes first conduct key discovery on each other to identify which of their neighbours has a key and which does not. To conduct key discovery, each key provided with a short

identification number before deploying it. Every node will publish its set of IDs of keys. This technique is also known as the E-G scheme. Connectivity of network is probability-based given by

$p=1-$  (probability of node not sharing a key).

$$p = 1 - \frac{k!(P-k)!(P-k)!}{P!k!(P-2k)!} \quad (2.1)$$

In equation 2.1, the probability that two nodes share the keys is given. In this equation, P is a big pool of keys, out of which k keys are randomly selected and stored in the sensor node. After deployment, every node checks whether it has a shared key with the neighbor node or not. For good connectivity of the graph, the value of p must be high. According to Chan et al. [16], a new improvement of the fundamental technique in which two nodes must exchange at least q keys to generate a pair-wise key was published in 2003, and it is still in use today. This increases system resilience since it is more difficult to compromise a q key than compromise a single key. The downside of this method is that all keys are stored on the node, requiring more storage space. This approach was proposed as a result of the basic system being examined. Two nodes in the E-G system interact with one another via the use of only one common key. The following method can be used for the removal of this problem. Create a common key for two nodes. Search for and identify all viable independent routes (for example, S1→S3→ S2, S1→S4→S5→S2), also known as high pathways. Create a set of n random numbers and distribute them evenly to all of the separate pathways. Node S2 generates a new key as

$$K' = K \oplus g_1 \oplus g_2 \oplus \dots \oplus g_n \quad (2.2)$$

The benefit of this system is the increased resilience obtained by surrendering transmission costs. In the equation, 2.2 K' is the final key calculated by the node by summing random numbers of nodes in different paths.

In 2004, Du, Wenliang, and colleagues [53] designed and deployed a pre-key distribution system that implements keys based on node geographical data. A sensor is deployed in a random manner. Because nodes from different groups need not communicate directly, and different key pools are used to allow communication between them. This improves the system's resilience and ability to withstand collusion-

based node capture attempts. The major drawback of this method is that it needs sensor node placement information, which is not always accessible, particularly in random deployment. This system beats the basic and q-composite systems because it requires less key maintenance on each node.

### 2.3 Matrix Based Scheme

For the purpose of dealing with scalability limitations, matrix-based techniques are proposed, which come at the cost of computation. Decomposition is the foundation of several matrix-based methods, including LU matrix decomposition, which uses a symmetric matrix as the basis of its operation. Blom's method [54] is a process for creating symmetric keys that are symmetric in nature (SKGS). When communicating in a network, each node will communicate with any other node that has less information than it. This kind of secret sharing is referred to as a threshold approach. In other words, the secret has been split and distributed across the nodes of the network. Attackers get access to the whole network after a specific number of nodes has been taken over, i.e., when a certain threshold has been reached by the attacker.

Blom's Scheme makes use of two matrices over  $GF(q)$ , a public  $(\lambda+1)n$  matrix  $M$  and a secret  $(\lambda + 1) (\lambda + 1)$  symmetric random matrix  $D$  that is only known to the KDC. Blom's Scheme is based on the MDS3 matrix  $M$ . In order to construct the symmetric  $n \times n$  matrix  $K = (DM)^T M$ , an element of which  $K_{i,j} = K_{j,i}$  corresponds to the key between nodes  $i$  and  $j$ , these matrices must first be calculated as shown in figure 2.2.

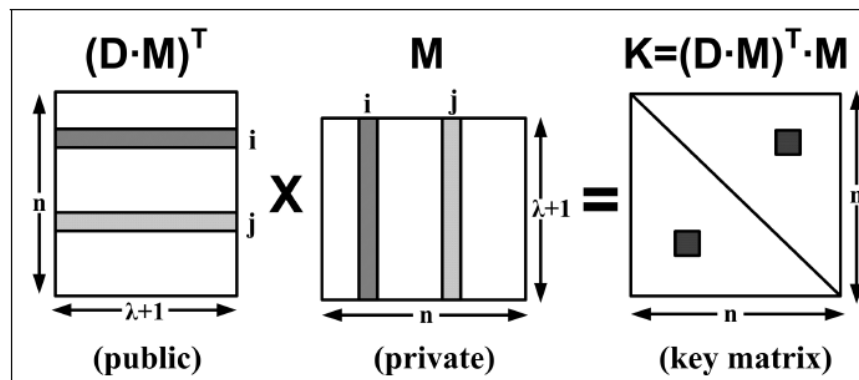


Fig 2.2. Matrix Key Calculation

Chang et al., 2005 [55] present a technique for distributed sensor networks based on LU decomposition. This approach also makes use of all three pre-deployment stages. If node A initiates communication with node B by sending its  $U$  matrix column, node B

times the value of that column by its row and transmits the hash of that number along with its column. Node A computes and determines the hash of that value, at which point the communication key is generated. If the hash values of both devices match, they will start talking through  $K_{AB}$ .

## **2.4 Location-Based Scheme**

In 2006, Mohamed F. Younis developed the location-aware system, which takes advantage of sensor location awareness to provide real-time information. But the authors came up with another key management system, which they dubbed the SHELL key management scheme (which stands for "Scalable, Hierarchical, Efficient, Location-aware, and Light-weight") [56]. Using sensor nodes with different capabilities, the researchers investigated a heterogeneous network. It is a system that works on the Exclusion Basis System of classification. This technique makes use of a key pool with a size of  $k + m$  keys, and a set of  $k$  keys is chosen from the aforementioned pool of keys to be loaded into each node. Even though key refresh processes in EBS-based systems are straightforward, they are very susceptible to assaults on collusion. Even though another scheme like SHELL contains a method to mitigate collision assaults, it still has several drawbacks to consider. In the first place, the storage requirement per node is significant due to the large number of keys that each node must retain; in the second, it places the responsibility of rekeying on key generation gateways; as a result, a breach of key generation gateway security will compromise network security. In addition, its functioning is tough to comprehend. When examining the architecture of a hierarchical network, this approach is inefficient in terms of storage since each node must keep various credentials.

In 2007, Xiaojiang Du et al. [57] proposed an effective key management approach for heterogeneous sensor networks that use high-end sensors with high reliability. The performance assessment and security analysis indicate that the key management scheme offers greater security while needing less complexity and a substantial decrease in storage requirements when compared to current key management schemes. Sensor networks used in the military, homeland security, and other hazardous situations must be secure. Previous sensor network security research has mostly focused on homogenous sensor networks. This is changing. The performance and scalability of

homogeneous ad hoc networks, according to studies, are both poor. Apart from that, many security techniques for homogeneously networked sensor systems are too expensive to implement due to the high communication costs, the high computing overhead, and/or the high storage requirements. Recent implementations of sensor network systems are increasingly incorporating heterogeneous architectures into their design principles. It is necessary to conduct additional security operations to use key management, a cryptographic primitive.

When Ling, D., and colleagues [4] proposed utilizing sensors' predicted locations to assist in the pre-distribution of keying materials in 2008, it was a ground-breaking idea. However, it is relatively difficult, though not impossible, to ensure that the anticipated positions of sensors are known ahead of time. This article proposes that sensor nodes be organized into groups and that nodes within the same group be positioned near to each other after deployment to eliminate reliance on projected placements and to create a realistic deployment paradigm. The article proposes a new cluster connected with other individuals' frameworks that may be utilized in combination with any current key pre-distribution method based on this idea. This paradigm differs from others in that it does not need previous knowledge of sensor anticipated locations, significantly simplifying the design of sensor networks. Furthermore, the study demonstrates that the framework may significantly enhance the speed and reliability of current key pre-distribution methods.

A common assumption in the majority of distributed key management systems, according to Lu, K. et al. [58], is that all sensor nodes have the same capabilities. Recent research, on the other hand, showed that by increasing the energy capacity and transmission capacities of a limited number of sensor nodes, the connectivity and lifetime of the sensor network might be substantially increased. Therefore, taking advantage of these diverse features to construct a viable distributed key management system has emerged as a critical issue that must be studied further. A framework for key management techniques in distributed wireless sensor networks with heterogeneous sensor nodes is presented by the authors in this paper, which is available online. A wireless sensor network with a small number of heterogeneous nodes, as shown by simulations, is more likely to achieve higher key connection and durability.

A Two-level Key Pool Design-based Random Key Pre-distribution is proposed in [8]

by Mohaisen A. et al., which consists of two main phases: offline and online distribution. During the offline phase, which an administrator completes before deployment, different keys are assigned to different network nodes. Secure communication between two nodes is established during the online phase. The two nodes identify a common shared key or establish a key route via one or more intermediate nodes. According to this paper, the author has re-examined random key distribution and establishment in the EG scheme, as well as re-designing the large key pool, generating smaller pools, and assigning randomly generated keys for each node from different and randomly selected sub-pools to achieve greater efficiency with less communication overhead while maintaining the same level of memory overhead for keys and reduplication keys. Although this method reduces transmission costs, it necessitates the use of additional processing and storage.

## 2.5 Polynomial Based Scheme

The first polynomial-based scheme was proposed by Blundo et al. in 1992 [59]. In this scheme, the author used randomly generated  $t$  degree polynomial

$$f(p, q) = \sum_{i,j=0}^t a_{ij} p^i q^j \quad (2.3)$$

over  $GF(q)$ , such that  $q$  is a prime number. Since the polynomial is symmetric  $f(p, q) = f(q, p)$ , the communication keys computed should be within  $q$ . Each node shares a polynomial share of  $f(i, q)$ . In equation 2.3  $a_{ij}$  is a constant and  $p$  and  $q$  are the secrets stored in the node.

Dai and Xu [60] solved the issue using a combination of LU decomposition and a polynomial pool-based approach in 2010. A polynomial pool is created, and the polynomial pool is used to construct an  $L$  matrix. This is an alternate technique for constructing the  $L$  matrix by establishing the key pool over  $GF(q)$ . Finally, the  $U$  matrix is based on the assumption that the product of  $L$  and  $U$  is the same as the symmetric matrix.

A. K. Das and colleagues proposed a random key distribution technique for large-scale distributed sensor networks in 2011 [61]. To preserve the necessary unpredictability in the key selection, this approach continuously establishes a connection between the ids

of neighbouring nodes and the keys held by those nodes. Our suggested approach outperforms current random key pre-distribution strategies in terms of node capture security. Furthermore, it provides a better mix of communication cost, network connectivity, and node takeover security than current random key pre-distribution methods. Furthermore, it readily enables dynamic node addition once the initial deployment of network nodes is complete.

MAKE [62], a modular arithmetic-based key management technique for cluster-based networks, was developed by M. Du and colleagues in 2012. The authors proposed a key management system that handles keys using the coherence of mathematical logic feature. In order for this system to work, each sensor node just has to keep a secret seed. This key seed is used to generate two different kinds of keys for each node. The cluster head's key, which is shared by all nodes, is the first. The group key is the second kind of key, and it is shared by all nodes in a cluster. The researchers propose a technique for upgrading the cluster's key seeds when the cluster head finds a corrupted member in its cluster. Each update message sent by the cluster head to a non-compromised member is encrypted using a surround alongside the group head and the matching member sensor node. According to this technique, breaching the secret of only one cluster head compromises the secrecy of all cluster nodes in a chain reaction manner.

## **2.6 Hash Key Chain Based**

Bechkit et al. (2013) [63] described a hash-based key pre-distribution method for WSN. An attacker can't see the keys that have been previously disseminated because of the hash algorithm. It is shown that by hiding keys via an effective hash chaining method, this strategy increases resistance to node capture and improves resilience to node capture. This class may be used for any pre-distribution scheme that is based on a key pool. In the beginning, the author used this approach to the well-known probabilistic q-composite scheme and then to the deterministic Symmetric Balanced Incomplete Block Design method, which resulted in a successful outcome. Each of the two distinct blocks is connected to the other by a single common key. After deployment, neighbouring sensor nodes share key IDs with one another. Alternatively, if they share one or more common keys, the pairwise key is calculated by hashing all of the shared keys together. In every other case, sensor nodes in close proximity should work together to find a safe route for generating a pair-wise key. This method is scalable, but it



requires a significant amount of memory to keep the key blocks and the key IDs that are assigned to them. The fact that an attacker may frequently compromise nodes is another disadvantage of this technique. As a result, network resilience quickly degrades. Du and Xiao et al. presented a new routing-driven key management method in [62] that only created shared keys for neighbours sensors that could interact with each other. In order to offer a reliable key management mechanism for HSN, elliptic curve cryptography was used. H Sensors are required to save all of the public keys of L Sensors, and L Sensors are required to save all of the public keys of H Sensors. A significant quantity of storage capacity and a large amount of energy is used by each node.

In 2015 Anita et al. [64] proposed a complex key management method for enabling secure communication across sensor nodes that blends the polynomial technique and q-composite scheme. The triple key arrangement is the underlying concept of the scheme. A triple key is created between communication nodes by utilizing intermediary nodes that serve as mediator nodes, increasing network resilience to node capture attempts. When a triple key is formed, it is a method of securing a group of three or more players in a game. However, since each node must store both the keyring and the polynomials, the amount of storage needed per node is significant.

M. L. Messai et al. (2015) presented a sequence-based insubstantial scheme for distributed networks in [65]. The first term and the recursive formula of a numerical series are fed into the sensor nodes in this system at the start. Following deployment, each pair of sensor nodes uses the two pieces of data to create a common key that they may all utilize. Sequences do not include arithmetic, geometric, and convergent sequences, among others. Because each node only has to keep two little bits of information, SKM is very space-efficient.

Zhang Y. et al. [66] presented a hybrid method in 2016 based on basic random key distribution and tree-based route key creation. An improved hybrid key management approach for WSNs with MS is presented in this paper, which incorporates both a Polynomial Pool-based key pre-distribution method and a Basic Random key pre-distribution method (PPBR). The method makes extensive use of these two kinds of techniques in order to enhance the difficulty of cracking the key system's security measures. It is also possible to significantly enhance the efficacy of storage and the

resilience of networks. For the purpose of effectively addressing the problem of communication link connection, the tree-based route key generation method has been suggested. When it comes to network resilience, connection, and storage effectiveness, the simulation clearly shows that the proposed system surpasses current widely used methods.

In addition, Zhang Y. et al. presented a hybrid key establishment method in [66]. It is a basic service in wireless sensor networks that involves establishing pair-wise keys for each pair of adjacent sensor nodes. This service serves as the basis for further security services such as authentication and encryption in wireless sensor networks (WSNs). However, since sensor nodes have limited energy, memory, and computational capabilities, it is difficult to establish the pair-wise key. Using a combination of polynomial pool-based and probabilistic key pre-distribution methods, this article presents a method for key pre-distribution in wireless sensor networks. A portion of the sensor nodes in the proposed system is pre-loaded with polynomial shares, and the polynomial shares are used to compute the keys that are used to form a key pool of the system's keys. Furthermore, the keys selected from this key pool are preloaded on the remaining sensor nodes in the network. It is determined how well the proposed system performs in terms of connection, attack resistance, memory consumption, and communication overhead. The simulation findings also show that the suggested strategy surpasses the existing methods of network resistance against node capture, which is encouraging. Security elements like authentication are also important since they allow authorized users to get access to data that has been made available by sensor nodes. Among those who have contributed to this work are Wu et al. [67].

In 2016, Bajestani M. F. et al. [68] presented an attack probability-based method against storage-bounded adversaries. The distribution area is split into zones with variable attack likelihood in this system. It is presumed that the adversary's storage capacity is limited and that not all messages can be kept. This method is very resistant to node breaches, and the attacker has a very minimal chance of finding the key in the event of eavesdropping. Key setup is a difficult issue in wireless sensor networks. Scalability and energy efficiency are important characteristics to look for in a key establishment system. In addition to being scalable, we present a method that uses resources based on assault likelihood within each area. The distribution area is split into zones with variable

attack likelihood in this system. It is presumed that the adversary's storage capacity is limited and that not all messages can be kept. This method is very resistant to node breaches, and the attacker has a very minimal chance of finding the key in the event of eavesdropping. The findings of the probability and simulation analyses indicate that the scheme offers the desired efficiency and that the amount of energy used by the system stays constant regardless of network size changes.

Choi, J. et al. [69] presented location-based key management robust against insider attacks in wireless sensor networks in 2017. Sensor nodes must create secret shared keys with neighbouring nodes to enable secure communications in wireless sensor networks (WSNs). Furthermore, the keys must be updated by overcoming insider threats posed by corrupted nodes. The author proposes a method for WSNs that takes into account insider risks. We chose location-dependent key management (LDK) as a viable scheme for our research after evaluating current location-based key management schemes and studying their benefits and drawbacks. We developed a novel key revision procedure that integrates grid-based location information to address a communication interference issue in LDK and related techniques. We also offer a key setup procedure that makes use of grid information. To successfully fight inside attackers, key update and revocation procedures must be used. When the minimal number of common keys needed for key formation is large, the scheme demonstrated that the technique might improve connection while reducing the compromise ratio. When an insider threat compromised a node, we could effectively rekey every SN except the damaged node using our technique. Finally, the hexagonal placement of anchor nodes has the potential to decrease network expenses.

Gandino F. et al. [70] presented a q-s composite based on the random prior distribution of the secret content in 2017. The primary advantage of the scheme is efficient memory management, which allows for the storage of a greater number of keys and, as a result, may enhance the protocol's robustness. This method also includes an upper limit on the number of beginning keys utilized to create a pair-wise key (called  $s$ ). Furthermore, rather than creating a pair-wise key for each neighbouring node, each node saves the information for key creation and computes the pair-wise key when the network's security mechanism requires it. A light key generation method based on the bitwise XOR operation is presented to avoid extra computation cost. In contrast to the restricted

resources of wireless sensor networks, this method has significant memory overheads. It leverages the best characteristics of random pre-distribution to enhance it with reduced needs.

Ahlawat P. et al., 2018 [71] propose an attack-resistant key pre-distribution method that takes adversary behaviour into account. The defender/network designer may effectively and efficiently build various countermeasures against hostile behaviour by understanding it. The author analyses the node capture attack issue and offers a safe hybrid key pre-distribution method (HKP-HD). The robustness of the q-composite system is combined with the threshold resistant polynomial technique in this approach. The suggested method seeks to strengthen the network's resistance against node capture assaults. The adversary is considered clever, with the intent of constructing an attack matrix by exploiting various weaknesses in the network. The attack matrix attempts to destroy the whole network with the fewest number of nodes. As a countermeasure, the network designer creates a comparable attack matrix based on the network's weaknesses, with sink as a key influencing element. The attack coefficient is calculated using this matrix.

In 2018 Ying Z. et al. [72] proposed schemes that consider the probability of node capture as a parameter. In this paper, the author proposed a model regarding the probability of nodes being captured which not only considers the energy factor but also considers the node capture probability when choosing a cluster head, so the node which holds a smaller capture probability tends to be cluster head. On the same day in the same year, Messai M. L. and colleagues [73] present a new symmetric key management technique for hierarchical WSNs, which allows for secure cluster construction (HWSNs). EAHKM+ is the new software (energy-aware hierarchical key management in WSNs).

Each sensor node gets just three keys before deployment. EAHKM+ ensures the establishment of a broadcast key between each sensor node and its cluster head. Most of the schemes make the network secure but do not take the consideration of memory requirements and computation requirements. The scheme is also lacking in providing scalability. In modern days the use of sensors is also changed as they communicate with different types of devices as in IoT, so key management research has the scope of heterogeneous networks. There is a trade-off between security and computation

required, so a key management scheme must provide adequate security as well as computation efficiency. Many schemes are prone to single point of failure problems due to their centralized nature. The energy requirement of these schemes is high as security is implemented in networks. Network lifetime can be improved by creating groups. A clustered-based network improves the lifetime of the network and gives the facility to implement a good key management scheme as the group head can work as a key distribution center.

In 2019 a new scheme for heterogeneous wireless sensor networks was given by Athmani, S. et al. [74], known as EDAK. EDAK provides a single lightweight protocol that may be used for both authentication and key distribution. The suggested methodology generates dynamic keys using pre-existing information. There is no need for a secure channel or a sharing phase with EDAK.

In [75] authors present a key scheme based on secure watchdog selection. Using key-based node identity verification and trust assessment methodologies, the proposed PKCMWS chooses numerous concealed watchdogs in clustered hierarchical wireless sensor networks. The proposed PKCMWS principles are contrasted with the existing immune system-based intrusion detection, effective, secure neighbour coverage system, spy mom, and Secure Reputation-Based Monitoring System (SRMS), which are other works related to this scheme. This technique finds issues with WSN watchdog-based IDS selection methods and offers novel secure watchdog selection algorithms in WSN to increase the watchdog availability ratio in the face of different types of attacks. The primary focus of this research is to develop a firmly secure key management scheme for large-scale wireless sensor networks. This research focuses on robust against node capture attacks and good connectivity of randomly deployed networks. This scheme generates the pairwise key for random networks, and these generated keys can also be used to create clusters securely. When clusters are made, the cluster head can start a group key and securely distribute the key to group members using a pairwise key. The first two research objects focus on generating pairwise keys for location-independent static networks. The third objective focuses on whether the cluster is created of these fixed homogenous networks.

Table 2.1. Comparison on the basis of different parameters

Scheme	Connectivity	Resiliency (Security)	Storage efficient	Computation	Rekeying Support	Scalable	Add delete new Node
Eschenauer and Gligor[52] (2002)	Average	Average	Average	Low	No	No	no
Chan et al. (2003)[41]	Poor	Good	No	High	No	No	no
Du et al. (2004)[53]	Depend on location	Good	Good	High	No	yes	manually
Du et al. (2007)[17]	Average	Low	Good	average	No	yes	yes
Ling et al. (2008)[76]	Depend on location	Good	Good	High	No	yes	manually
Mohaisen et al. (2010)[8]	Low	average	No	High	No	No	offline
Bechkit et al. (2013)[63]	Good	Good	Good	High	No	No	no
Zhang et al. (2016)[66]	Very good	Low	Good	High	Yes	No	yes
Ahlawat P. et al. HKP-HD(2018)[71]	Poor	Good	No	High	No	yes	no
Messai M. et al. EAHKM+(2018)[77]	Good	Good	No	High	No	yes	no
Ying Zhang(2018)[72]	Good	Good	No	High	No	No	yes
Athmani, S., Bilami, A. (2019) EDAK [74]	Good	Low	Yes	High	No	No	No
K Hamsha, GS Nagaraja 2019 [78]	Low	Good	Low	Yes	No	No	No
MS Yousefpoor, H Barati 2020 DSKMS [79]	Good	Low	Low	High	No	No	Yes
Rajasoundaran, S., et al. (2021) PKCMWS [75]	Good	Average	Low	High	Yes	Yes	Yes
P Alimoradi, A Barati 2021 [80]	Good	Good	Low	High	Yes	Yes	Yes

A dynamic network node may leave the group and re-join the new group, but in this case, random deployment in the isolated areas goes out of the research. Nodes deployment in the hostile area is the research's primary assumption and model structure. So if the same random network is converted to clustered network to save energy, the scheme works as well. The last objective is to implement the strategy in modern networks where no constraints of power but heterogeneous nature of devices are there. So in the scheme's aim, take advantage of the controller and use the controller as KDC and authentication identity. The same problems can be found in large-scale mobile networks and modern IoT networks. Heterogeneous types of networks use systemic deployment and may be location-aware. The cluster head has high power and is located in the middle of members groups. Further research for the complex mobile network may be continued, and a modified scheme may be used for IoT networks.

## **Chapter 3**

### **Problem Identification and Objectives**

#### **3.1 State of Art in Key Management**

In secure communication systems, key setup techniques are at the core of the system's operation. Key management systems establish secure connections between sensors on neighbouring networks during the network construction phase. For wireless sensor networks, there are two phases that must be taken into consideration, start-up and network configuration. There are two phases to the network construction process: the finding of shared keys and the configuration of keys. Two sensors work together to locate a common key in a shared key finding to unlock the door. It is necessary to create a shared key between two sensors to communicate securely with one another [81]. Mature wired and wireless networks use public-key cryptography to protect data transmissions. It comprises two parts, an individual's secret key and a public key that uses the same secret key. Aside from that, it is impossible to deduce the private key from the public key. The usage of big keys (in comparison to symmetric keys) and complicated encryption and decryption mathematics are required to achieve this objective, and this is a critical point to remember [82]. Symmetric key cryptography makes use of the same key for both encryption and decryption, allowing for smaller keys and less complicated encryption arithmetic to be used. However, since the same key is used for encrypting and decrypting activities, the administration of these keys is much more important than the management of public-key cryptography keys [83]. Because wireless sensor networks (WSNs) lack the necessary storage and computing capabilities to employ public-key encryption, symmetric-key cryptography must be utilized. To ensure that wireless sensor networks (WSNs) last as long as possible, we investigate the necessity for an administrator key management layer to be added to current session key management systems. Popular key management systems assume that global or static administrator keys are adequate to re-configure a network or encrypt session keys, which is incorrect. To make things easier, session key management systems make the simplifying assumption that these administration keys cannot be compromised since they are either seldom used or are kept in tamper-resistant hardware. According to these assumptions, the network's lifespan is limited to the



amount of time it takes for the static administrator keys to be compromised. We discovered the necessity to investigate a two-tiered key management system with dynamic session keys that allow network re-configurations to support long-lived WSNs without making any assumptions about the future.

To develop a specific key method that can only be utilized for all kinds of networks with a variety of topologies is almost impossible. In dispersed networks, a number of different key methods are used. There are many effective management systems available that satisfy the vast majority of the criteria [84]. Instead of focusing only on fulfilling all of the security requirements, a system must consider the limited resources of networks. The computation is carried out by the sensor node. As a consequence, there should be no plan put into effect. In terms of storage and processing, it is very resource-intensive. It is not acceptable to do so. Invest all of your time and energy into the preliminary preparations. There is another need for scalability, which ensures that the technique can be utilized for networks with a variety of sensor sizes, extending between 100 to 1000 and maybe even 10,000 sensors.

The performance of this technique will be excellent if you are working with sensors that have restricted capabilities. Both the availability of resources and the capacity to scale up are significant factors in the decision-making process. The truth is that they are much less effective when it comes to overall performance than they were previously thought of [85]. When a separate key is used for each pair of objects, on the other hand, if there are no nodes, then this method is very safe since any key can't compromise it. It does not impact other connections, but it is not scalable and is not feasible to implement.

The primary goal of key management is to offer safe methods for managing encryption key exchange secure information between the nodes. After conducting an analytical study on random pre-distribution, it found that most current research focused on providing security without consideration of computation and storage space. Many key management schemes were proposed, but none of them define security requirements for Morden applications that include IoT. In new applications like smart homes health monitoring systems, there are different types and ranges of devices interacting with each other, so key management should be applicable to heterogeneous and scalable with consideration of limited storage and computation. Protocols define communication

sequences like message transfers and computation steps that are used to distribute keys.

### 3.2 Research Gap

Key management systems are used to keep the information confidential. Furthermore, keys may help verify real network nodes. An attacker may attempt to break the secret key of communication nodes to get secret information. The authentication key may be attacked and compromised to gain control over a node. Attackers attempt to break a secret key by analyzing communication patterns in order to deduce the secret key. They even attempt to record certain encrypted talks with the aim of replaying them later. Maintaining keys at appropriate time intervals is essential for avoiding attackers from guessing secret keys and jeopardizing data security.

After conducting an analytical study on the key management schemes, it has been found that most current research focuses on providing security without consideration of computation and storage space. Some of the main deficiency found in existing schemes are as follow

- Many schemes like matrix-based are not scalable as they require information of the number of nodes at the start before the deployment. Existing methods do not allow the addition of new nodes in the deployed sensor networks. In the literature review, we see that the proposed polynomial-based schemes required a very high computation.
- Some schemes like SHELL required location information in the networks, which is generally unavailable in randomly deployed sensor networks.
- The proposed schemes in which the base station works as KDC are suffering from high traffic, and most of the energy of the sensor is consumed by getting the key from the KDC. These schemes also suffered from single-point failure.
- In randomly deployed large networks, connectivity is a big problem as nodes may not find the pairwise key and are not be able to send the secure message to each other. The Basic EG scheme is given for large sensor networks for pairwise keys, but it does not support ket revocation.
- Most of the existing key schemes do not support key revocation and key refresh at a regular interval which is the main requirement in today's dynamic environment.

- The existing schemes are also not very efficient against node capture attacks and do not maintain the system's resiliency. The q-composite and matrix-based scheme system loosed all the keys if a portion of the system is compromised.
- Storage and energy requirements are also problems in some schemes.

### **3.3 Objectives of Research**

1.1 Design a scalable and storage efficient secure key management approach for static wireless sensor networks by using the hash chain-based key along with re-keying support to refresh the secret keys whenever required.

1.2 A mathematical model is created to study and analysis of

- a. Connectivity of network in probabilistic scheme
- b. Resiliency against node capture in random pair wise key

2. The design of a Scalable Dynamic Keying Technique for authentication in clustered LEACH like protocol for wireless sensor networks that support rekeying and scalability through adding new node to existing network.

3. Propose a complete security framework with dynamic key management for modern applications like smart building that include different types of heterogeneous devices, using random key pre-distribution.

### **3.4 Proposed Methodology**

When it comes to mission-critical applications, wireless sensor networks are becoming a more popular and obvious choice. It is possible that the ability of the WSNs to carry out their duties may be compromised in the event of an unforeseen occurrence. The implications of these decisions must be carefully considered both before and during deployment in order to obtain an acceptable level of perceived performance while also avoiding potentially dangerous unanticipated consequences during the operation, as described above. This method, which uses a formal verification strategy based on occurrences, is intended to assess and enhance the dependability level of WSNs via the use of formal verification events. In this research, we can use various simulators to generate the result with various parameters and compare the existing techniques. It is known that it is not possible to have a totally secure network. However, it is possible to control the risks through appropriate methodologies establishing security levels and

periodic evaluations. The flow chart of the proposed methodology is shown in figure 3.1.

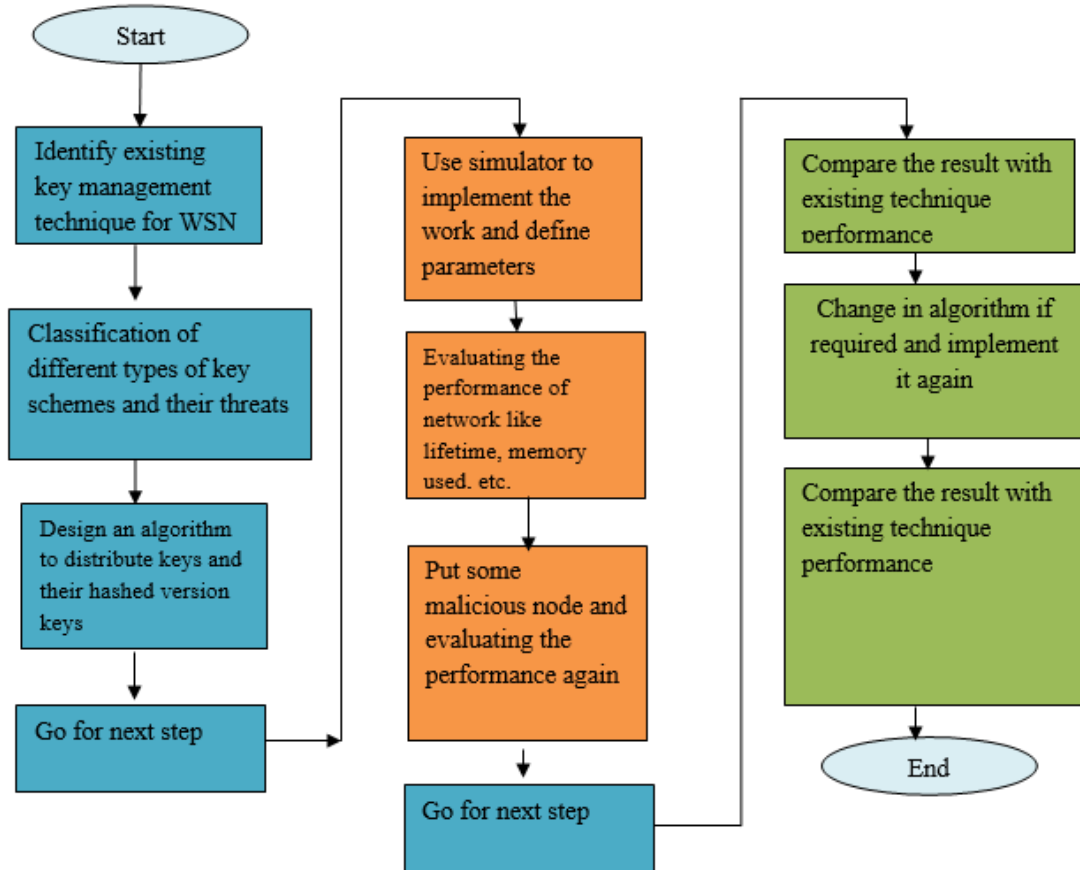


Fig 3.1. Flowchart of the proposed methodology

We can configure a simulator with different parameters and collect the results for the same. Along with the simulator result, a mathematical model is also prepared for probabilistic-based schemes. From various mathematical equations, we can find the probability of various events. In a mathematical model, we can use simple mathematical formula such as if the probability of any event is  $p$ , then the probability of not happening that event is

$$P'=1-P \quad (3.1)$$

We attach the key to every node and find the probability of connectivity of the network. The same parameters are applied to the simulator and run the simulation many times. Calculate the time of message sending, receiving, and processing. The primary focus of this research is to develop a firmly secure key management scheme for large-scale

wireless sensor networks. This research focuses on robust against node capture attacks and good connectivity of randomly deployed networks. This scheme generates the pairwise key for random networks, and these generated keys can also be used to create clusters securely. When clusters are made, the cluster head can start a group key and securely distribute the key to group members using a pairwise key. The first two objects of research focus on generating pairwise keys for location-independent static networks. The third objective focuses on whether the cluster is created of these fixed homogenous networks.

## Chapter 4

### Scalable and Storage Efficient Key Scheme

#### 4.1 Introduction

Sensor nodes have a limited amount of memory, computing power, connection, and energy to operate at peak performance. When attempting to resolve an issue in this area, it is important to keep in mind the resources utilized in the process. The same concept applies to the management of keys [86]. Generally speaking, this is a trade-off determined by the particular application scenario. The overwhelming majority of key management systems mentioned in the literature are completely incapable of being scaled up or down. This chapter describes a new key management technique for a network of sensors that are deployed randomly. For example, you might use this technique to generate paired keys for your network, which you could use to allow safe clustering. A static node is a network model component that specifies the structure of the network model. The nodes do not move from their original locations once placed in the field with all of the nodes. Before deployment, each node is pre-loaded with a limited quantity of sensitive information accessible to authorized users. Because of the deployment, nodes can recognize and communicate with their neighbors. Setting up and renewing the keys are done after deployment.

#### 4.2 Network Model

WSN models with the following characteristics are anticipated to exist and are utilized in this study. Every sensor node has its own identity, and every sensor device shares equal memory and processing power as the base station, as well as the same speed and sensing capabilities [87]. Base stations are more capable than other kinds of stations and can communicate over longer distances. The sensor node and the base station (BS) remain immobile and in an unknown position. Multi-hop communication may be used to communicate between two nodes as long as  $s_i$  can listen to  $s_j$  and  $s_j$  can listen to  $s_i$ . The attacker is presumed to be clever yet limited in their skills. Before gaining complete control of the network, the attacker seizes control of a node and hides behind it for a while. If a node's conduct is determined to be malicious, the key must be renewed, and the node must be removed from the networks. Table 4.1 contains notations for different network parameters.

Table 4.1. Notations

Notation	Description
$S_i$	$i$ th sensor node in the network denotes the unique id
BS	Base station
$CH_i$	$i$ th cluster head
$k_i$	$i$ th key chain in key pool
$k_{ij}$	$j$ time hash key from $i$ th key chain
$H_n$	No of time key is hashed $n$ times
$u_0$	First-term for recursive formula
$H_{u_0}(K)$	Key $K$ is hashed $u_0$ times

### 4.3 Proposed Scheme

We proposed and developed a decentralized approach for homogeneous cluster-based architecture. Some parameters like resiliency, single-point failure, and scalability were not incorporated in any previously proposed key management systems. Our system is decentralized, with no single point of failure. The scheme's resilience is also extreme since multiple nodes use a single set of keys. It also has the benefit of being scalable, energy-efficient, and uses low power. Our approach is dynamic, and it gets more detectable as the topology of the system evolves over time. It is possible to have almost limitless base station memory and processing performance with the suggested design. All of the nodes have the same kinds and have the same power supply.

### 4.4 Key Chain

To implement this method, we use a key chain, as shown in Figure 4.1 and explained in the article [63]. A key pool consisting of a  $P$  non-colliding hash chain of  $L$  length is maintained by the base station, and every value in a single chain is considered a possible key in this approach. In a chain of keys, the hash of the previous key is used to create the next key in the chain [88].

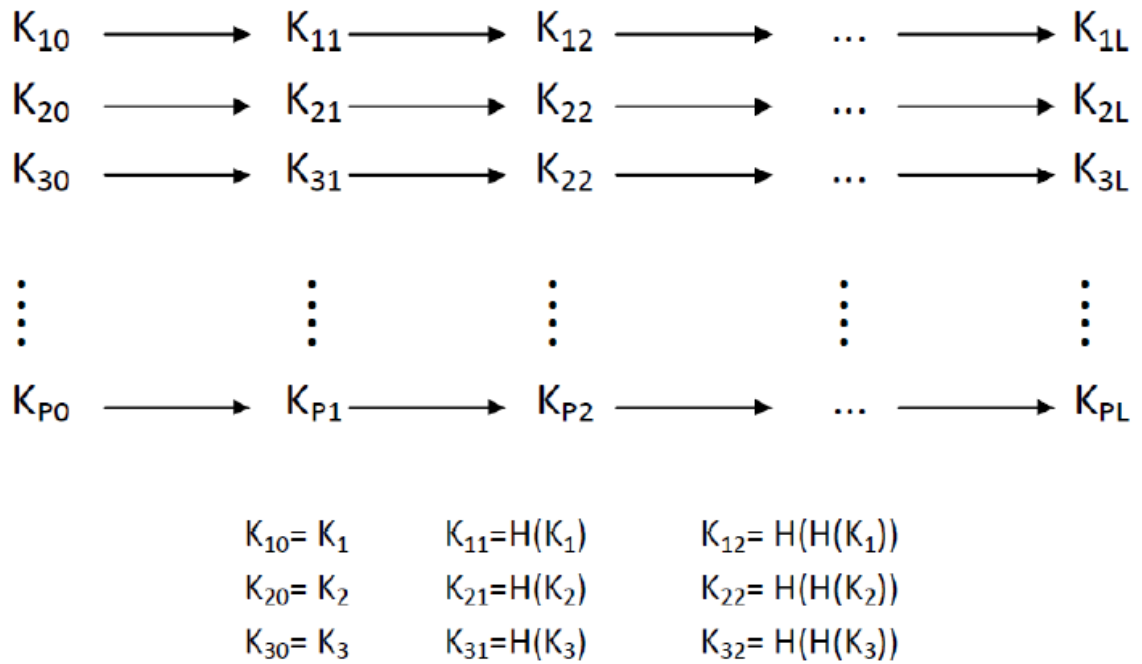


Fig. 4.1 Key Chain Pool

The network is built by selecting  $m$  chains at random and assigning  $m$  keys to each node before it is deployed. It is allowed to hash the key an unlimited number of times. A key can be hashed any number of times, and this number is known as key chain length. These three stages of the proposed scheme are as follows: key pre-distribution and cluster formation; pairwise key and cluster key generation; and key refresh after an interval on-demand. Communication between two nodes is possible when both nodes have access to the same set of secrets.

A sequence number is also used to renew the key or to create a new key for a new node when the key is no longer valid. It is essential to generate keys (for nodes that are unable to recognize a common key) via the use of sequence-based key generation methods for them to communicate. Each node also includes a seed value, which is sometimes known as the beginning term, as well as the mathematical formula for the series in question. In mathematics, numerical sequences, such as 1, 2, 3, and so on, represent lists of numbers that are generated for a series of integers. In mathematics, a discrete function relates a number, denoted  $u_n$ , with any integer  $n$ . To create the next word that will be used as a key, a recursive formula is used. Non-convergent series is neither mathematical nor geometric, and they are also known as non-convergent series. The values of the



sequence terms are complicated to derive for an attacker in the case of a non-convergent series where there is no convergence.

#### **4.5 Key Pre-distribution and Initialization of Nodes**

While establishing a sensor network, many procedures must be completed before and after the sensor network is enabled. A hash function and a hash chain key pool are created once this phase of the network is finished. An initial network phase is utilized to construct a non convergent recursive equation that will be employed later on in the procedure. For the sensor network to operate properly, it must initially be provided with three pieces of information: a hash chain, the first term of a series, and the first term of a recursive function. Before the network is randomly dispersed, these three bits of information are put in each network node. Even though the hash chain is used to assign paired keys to each node, owing to the use of key pre-distribution, it is a probabilistic method that does not ensure that every pair has a common key with the other pair. If a common key between two nodes cannot be discovered by comparing their sequence numbers and using the recursive method, one is generated. Node settings such as node IDs, transmission range, keying material, and any other required specifications are set up during this phase. During this phase, each node in the network is given a unique ID number. To make things even more complicated, each node is given a set of keys ( ${}^P C_m$ ) from the large key pool  $P$ , as well as the integer  $u_0$ , which indicates how many times each key will be hashed. As a bonus, the sensor also stores a recursive formula. After they've been activated, they'll be deployed at random throughout the target region, which they'll be responsible for monitoring.

#### **4.6 Key Setup**

After being generated at random, each node broadcasts its unique identifier and key chain identifier to all of its neighbors. It is possible to generate shared keys if the key chains in both nodes are identical. If the key chains do not match, the common key is created by combining the seed value and the stored function from the two key chains. The two nodes communicate with each other by transmitting their id and a random integer nonce in order to generate a shared key using the seed value and function. In order to guarantee message integrity, the hash value of the message is also sent. It is

necessary to take the hash of the message  $u_0$  times. This results in the message  $(S_i \parallel N_i \parallel H_{u_0}(S_i \parallel N_i))$  being sent by the node  $S_i$ , where  $N_i$  is the nonce generated by  $S_i$  and  $u_0$  is the first term of the non-convergent formula stored in  $S_i$ . The one-way hash function is denoted by the letter  $H$ . Following receipt of the message, Node  $S_j$  creates the secret key by following the procedures outlined below:

- Node  $S_i$  generate a nonce  $N_i$  and send the message  $\{S_i \parallel N_i \parallel H_{u_0}(S_i \parallel N_i)\}_{S_j}$
- Node  $S_j$  generate a nonce  $N_j$  and send the message  $\{S_j \parallel N_j \parallel H_{u_0}(S_j \parallel N_j)\}_{S_i}$
- After receiving the message  $S_i$  and  $S_j$  computes the  $u_{N_i}$  and  $u_{N_j}$  and
- generate the common key  $H(u_{N_i} \parallel u_{N_j} \parallel S_i \parallel S_j)$

After each pair of nodes has established a pairwise key, each node may interact with the others by sending messages using the pairwise key.  $H_u(K)$  indicates that  $K$  is hashed  $u$  times. It is the irreversible process used as a hash chain to improve security. A node may communicate with a base station using any key or combination of keys, and the message and key ID can be transmitted to the base station, which will utilize the key to decode the message. After the pairwise keys have been formed, all of the keys stored in the node are hashed again, simulating the effect of erasing the keys from memory. During this phase, nodes also choose the cluster head based on the node weight that is the most difficult to reach while using the least energy and traveling the shortest distance. Following the election of the CH, the group key or cluster key is generated and distributed to the network's member nodes [89].

#### 4.7 Key Renewal

When a cluster head is changed, the network's life expectancy is increased. When the network's topology is altered, or when the cluster head is changed, the network's life expectancy is increased. If a node is captured, we must modify the configuration of the node in order to isolate it. The keys are stored in the node that holds that particular key. It is possible to use a number of methods to update a member node's keys or to add a new node to the network. A cluster head may renew the key at any moment or in response to the cluster's needs. To do this, the CH must create a new group key and distribute it to all group nodes, which are then encoded using a pair of keys.

---

**Algorithm 4.1: Algorithm for Key Setup**

---

Input	:	Network
Step 1	:	Every node $S_i$ broadcasts a message $S_i \rightarrow * : M\{S_i, K_i\}$
Step 2	:	Upon receiving the $M S_j$ do the following: for $i$ and $j$ If $(K_i == K_j)$ list = list + $(S_i, K_i)$ else list = list + $(S_i, 0)$
Step 3	:	Every node $S_i$ stores the common key in the list else stores 0
Step 4	:	For every node $S_j$ where $K_j == 0$ Node $S_i$ send a message $S_i \rightarrow S_j: M\{S_i, U_i, N_i\}$
Step 5	:	Upon receiving $M S_j$ do the following If $(K_i != 0)$ If $(U_j < U_i)$ $H(U_j - U_i) K_i$ Else generate $x = H(U_{N_i}    U_{N_j}    S_i    S_j)$ list = list + $(S_i, x)$
Step 6	:	The node with the maximum degree is selected as CH
Step 7	:	CH generates a group key and distribute the members
Step 8	:	Every key stored in the node is hashed by one more time $K_i = H(K_i), U_0 = U_0 + 1$

---

#### 4.8 Node Addition

After a prolonged amount of time, a node's energy reserves may be depleted, and the node may cease to operate. It is necessary to either install new sensor nodes or replace any current sensor nodes that have failed. It is possible to add new sensor nodes to the network while simultaneously removing an old node while keeping the key associated with the new node using our approach. A newly added node may be able to share a common key with previously deployed nodes in the network as well as nodes in the newly added node's immediate proximity by using the SSEKMS. The key pool is updated when a new node is added to the network or an old node is removed from the network, for example. Before being deployed in the field, the new node is provided with a set of keys from the key pool. In addition, since the base station is aware of the node's id and the list of OK keys that have been loaded into the node, it may assign the

same key in the event of a node replacement or a new set of keys in the event of a new node addition. a. This node attempts to find a common key with another node in the vicinity. If a common key exist between two nodes then this pairwise key is used, otherwise the sequence number and function are used to construct the shared key with neighbors.

---

**Algorithm 4.2: Algorithm to maintain the keys of the newly added node**

---

Input	:	Network
Step 1	:	A set of m random keys from key pool S and loaded in the node
Step 2	:	Recursive function and first term $u_0$ is also stored and deploy
Step 3	:	The first node try to find a common key chain with neighbours by sending its id $S_i$ and key ids stored in this node
Step 4	:	if node find common key use that as shared key else use random number and function to generate the key
Step 5	:	Node choose the cluster head as per weightage of nodes

---

#### 4.9 Key Refresh

BS may initiate the key refresh phase if a node is hacked or its purpose changes or network topology changes. It is possible to renew the key for a single node or the whole network.

---

**Algorithm 4.3: Refresh the key of the node**

---

Input	:	Network
Step 1	:	BS initiates key refresh by broadcasting message $BS \rightarrow M\{ \text{Hello, BS, Level}=0, \text{Energy}=\infty \}$
Step 2	:	All the sensor nodes receive the message set $u_0=L$ and forward the message by increasing the level and setting their id and energy level.
Step 3	:	After setting up the value of $u_0$ all nodes execute algorithm 1
Step 4	:	Node chose the cluster head as per weight

---

As soon as a base station updates the keys of a specific node, the node receives an additional value of  $u_0$ , which it uses as a starting term to generate the new keys according to method 1. Sending a broadcast message to the network base station will kick off the key renewal process for all nodes in the network at the same time. The

level, energy, and id of the node are all included inside this message. After receiving a message, each node returns the initial value  $u_0$  to the node level and executes algorithm 1 to reset the key. Every time a key is refreshed, the value of the level is reset to the length of the key chain, which is the greatest length possible.

## Chapter 5

### Performance Evaluation and Simulation Result

#### 5.1 Performance Parameters

To be considered a secure protocol, all key management systems must meet some conventional security criteria. A similar argument may be made for dynamic key management systems. Several dynamic key management assessment criteria are highlighted depending on the system and application environment characteristics. The criteria for evaluating dynamic key management methods in wireless sensor networks are discussed in the next section. Simplicio et al. [90] divided their pre-distribution key management assessment criteria based on sensor nodes and networking limitations into three categories: security, efficiency, and flexibility. On the basis of categorization and the specific characteristics of dynamic key management, we offer fundamental criteria and assessment metrics for the scalable and refreshing key scheme. Network lifetime, energy consumed, and delay can be measured in simulating environment [91].

##### 5.1.1 Security Metrics

Hostile nodes inside a network access to cryptographic keys in order to carry out harmful activities [92]. The integrity of a compromised sensor node may be restored once the secret key has been revoked and a new secret key has been created for the sensor node [93]. You must ensure that all sensor nodes connected to a compromised node are notified of the breach, with the exception of those used by the attacker. Additional safeguards include the prevention of future communication between previously compromised nodes as well as the loss of the secret keys for forwarding and backward secrecy protection. During the deployment process, a resilient node will also resist being hijacked or duplicated at any time during the process.

- **Node's revocation:** The revocation of a node. Sensor nodes that have been compromised should be removed from the network as soon as they are discovered using a reliable manner. A hacked node may deviate from the network's behaviour by injecting fake data or altering data from trusted nodes, and node's revocation may be used to prevent this from happening.
- **Forward and Backward secrecy:** Confidentiality is maintained in both forward and backward directions. Forward secrecy is needed to prevent a node

from reusing an old key to decode fresh messages in the future. Backward secrecy is used in place of forwarding secrecy to prevent a node with a new key from traveling back in time to decode previously received messages encrypted with earlier keys using the new key [94]. The use of forwarding and reverse secrecy prevents node capture attempts from succeeding.

- **Collision Resistance:** When collusion happens, the capacity to identify and reject it. An attacker may launch an assault on the network by compromising multiple nodes and compelling them to collectively reveal all system keys, ultimately gaining control of the whole network. An efficient dynamic key establishment technique must survive the collision of newly joined and compromised nodes to work correctly.
- **Resiliency.** As a result of an adversary physically assaulting a sensor node to extract secret information from its memory, resilience is defined as the capacity to resist node capture. When a single seized node is examined, this technique assesses how the rest of the network is affected. When an attacker cannot affect any node other than the one taken, the resilience of a key management system is strong. As an alternative, if a single node is hacked, the whole network is affected, and the system's overall reliability and resilience are jeopardized.

### 5.1.2 Efficiency Metrics

The frequency of data transmission needed for rekeying, the overall quantity of cryptographic keys required, and the number of operations must all be maintained to a minimum degree that is practically feasible under the conditions. Cryptographic keys, on the other hand, should be reduced to a bare minimum to the extent that this is practically feasible. Random node sources of energy and memory space are no longer a constraint on the network's growth. When it comes to the following requirements, the dynamic key distribution should not put a significant load on energy sensors. The main efficiency metrics are

- **Memory.** Memory consumption is required in order to store security credentials such as keys such as public or private keys and pairwise keys, or a user certificate and IDs, as well as a certificate of adjacent nodes' reputation, among other things.

- **Bandwidth.** Node replenishment, node eviction, and the key creation process all need a large number of messages of different sizes to be sent back and forth.
- **Energy.** In the key setup process, data transmission and reception, and the computational technique for the generation and distribution of new keys, among other things, a considerable amount of energy is used daily.

### 5.1.3. Flexibility Metrics

It is important that the key setup methods be adaptable enough to work effectively in the diverse range of situations that WSN applications may encounter. The following are the essential flexibility metrics:

- **Mobility.** The vast majority of network designs are designed on the assumption that sensor nodes would stay stable. Certain applications, on the other hand, demand the flexibility of the core network, sensor network, or even both, while others do not [95]. Relocated nodes should be given new keys by the key establishment, allowing them to interact with their new neighbours due to the relocation. Key generation and distribution become more challenging when dealing with changing nodes because mobility capacity, in addition to energy and bandwidth, becomes a critical issue.
- **Scalability.** Depending on the intricacy of the sensor network, hundreds or even thousands of IoT devices may be present in the detecting region at any time. In addition, it should be remembered that nodes may join or depart the sensor network at any moment throughout its operational lifespan. Therefore, systems for dynamically key generation must be scalable so that they can handle a wide variety of network topologies, as previously stated. Meanwhile, when applied to bigger networks, performance must not degrade.
- **Key connectivity.** In cryptography, connectivity means shearing secret keys between two nodes and must be able send secret messages to each other. A local connection is defined as the connectedness between any two nodes that are in close proximity to one another in terms of distance. Global connection, on the other hand, refers to the interconnectedness of the whole network as a whole. It is critical to maintaining a strong key connection after each rekeying procedure in order to ensure security continuity.



It is possible to simulate a real-world network on one computer by creating scripts in C++ or Python, which are then run on the machine in question. A discrete event network simulator such as NS-3 is intended mainly for research and teaching purposes, as implied by the program's name. It also encourages the reuse of several real procedure deployments within NS-3, as demonstrated by the reuse of many existing real-world procedure deployments within NS-3, as well as a variety of other features. One of the key components of the NS-3 framework is a real-time scheduler, which enables various "simulation in the loop" scenarios to be implemented when dealing with real-world systems. The transmission and reception of packets produced by NS-3 on actual network devices are possible; in addition, the NS3 simulator has the potential to be utilized as a connection architecture to create link effects between virtual machines, for example connections between nodes may be established using NS3 in several ways, including point-to-point, wireless, CSMA, and so on. It is the same as a LAN connection between two computers when used as a point-to-point network connection. It is the same as a WiFi connection between different PCs and routers when referring to wireless connections. It is the same as having a bus topology between computers when using a CSMA connection. We attempt to install a network interface card (NIC) on each node once the connections have been established to allow network connectivity. NS-3 is an event-based network simulator that may be used to simulate various networks. Ns-3 is more in line with how real-world systems are developed than NS-2 since it has a lower abstraction level. The source code is available to the public under the GPLv2 license. C++ objects are used to represent simulation models. The sequence of events and the results of data collection is the emphasis. This proposes a simulation time point with a trigger and a start button for each occurrence. If you have a computer that runs NS3, you can do real-time calculations. If you're pretending to be on a network, you'll almost definitely need this if you're exchanging packets with a real-world host in the order that the event data is saved. The flow of data in NS3 between various components is shown in figure 5.1, and different modules available in NS3 are shown in figure 5.2. NS3 recommends Waf, a Python-based build system. The NS3 project's source code is documented using Doxygen. Nodes, applications, net devices, channels, and topological aids are all NS3 network simulation implementation instances. The fundamental features of an NS3 were established before moving on to

abstractions. Several apps for network-related tasks are included with the NS3. In the NS3 tutorial, there are several examples. A location in an NS3 is a point of contact, such as an end system or a router. All future activities will be built on this foundation. Nodes. Channels are used to link nodes together. They display data transfer differently. In NS3, two classes are defined: WifiChannel and PointToPointChannel. The phrase PointToPointChannel refers to the process of connecting two endpoints in a straightforward, direct, and connected manner. WifiChannel is a wireless internet connection in the form of a connector. The last and third most important internet devices are all abstractions. Their ties to nodes are what keep them connected. As a consequence, it is made up of a variety of communication routes. Internet devices exist in a variety of forms and sizes due to the variety of media sources available. One network may be connected to several others in the actual world. The software is current. Each NS3 simulation is a substantial abstraction. It comprises the nodes' real function and the simulation developer's ability to put it into action. The configuration and customization of these apps are important.

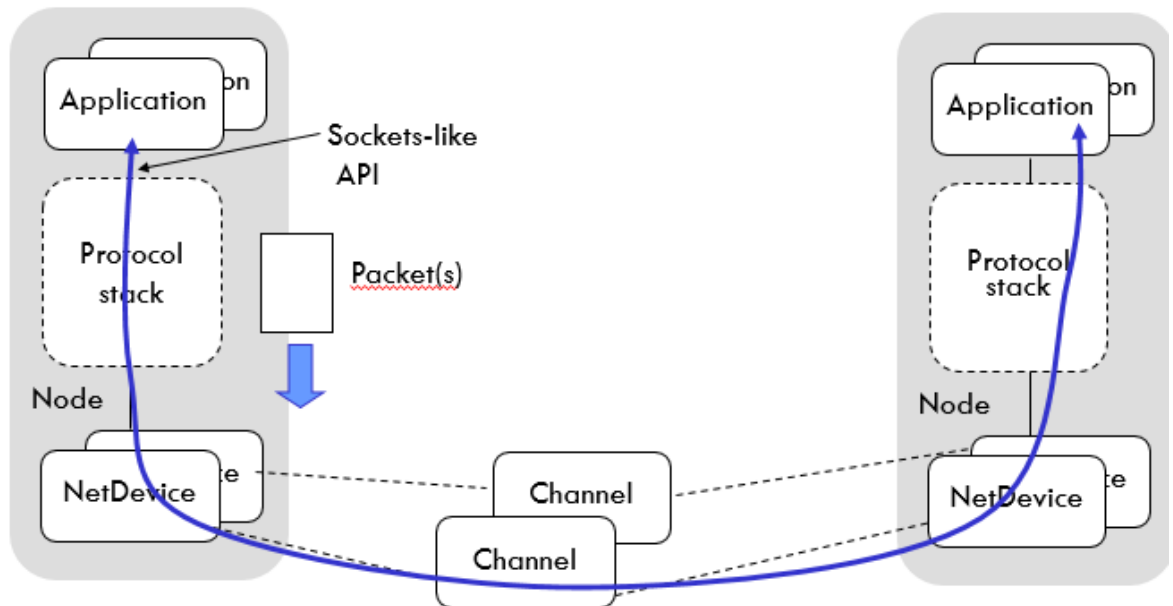


Fig. 5.1 Basic NS3 Data Flow Model

## 5.2 Performance Evaluation

To evaluate the different parameters of our system, we used statistical modeling and the simulation environment. At present, theoretical research is focusing on the features, connection, memory cost, and resistance to site acquisition of the network topology. A

network connection is established when network entities in reach of one another send a secure message to each other, signalling that they have a shared secret key [73]. A secure communication exchange uses these shared secrets called keys. Estimating your storage and computation overhead can assist you in determining how much storage and processing power you'll need to accomplish your activities. Furthermore, while assessing the impact of node capture, the notion of resilience must be taken into account.

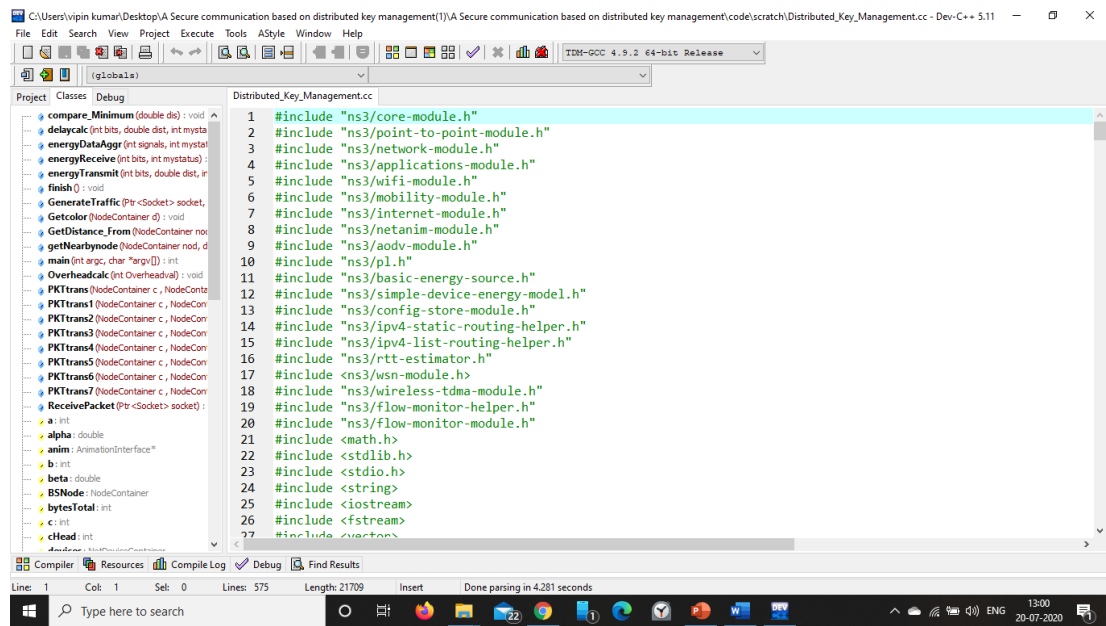


Fig. 5.2. NS3 Modules

Table 5.1 shows the comparison of different schemes with respect to different parameters. KPM is the Key Management Scheme given in [96], does not support node addition. DKMM is the dynamic key management method [97] that depends on location information, while EAHKM+ [73] energy-aware secure cluster scheme does not very efficient for memory saving. The proposed scheme provides all the solutions required for a good key scheme. Our proposed scheme, “scalable and storage efficient dynamic key management” SSEKMS is storage efficient and does not require any location information [69].

Table 5.1. Key Parameters

	<b>Key refresh</b>	<b>Node addition</b>	<b>Location based</b>	<b>Memory efficient</b>
KMP [96]	Yes	No	No	No
DKMM [97]	Yes	Yes	Yes	Yes
EAHKM+ [73]	Yes	Yes	No	No
SSEKMS(proposed)	Yes	Yes	No	Yes

### 5.3 Safety Analysis

It is also possible to use sensor networks in distant places and abandoned areas to detect and collect information about a wide range of variables. A consequence of this is that wireless sensor networks are more susceptible to a variety of attacks. Attacks like replay attacks and node replication are well-known in the security community [98]. This kind of attack is not successful against our system, and our system's resistance is verified against node capture. When a node is seized, resilience refers to the likelihood that the key will be revealed. The resilience of the node is reduced to zero, and the keys are destroyed if it is seized.

#### 5.3.1 Node Replication Attack

An attacker may install his own controlled sensor during node replication, interrupting network traffic. To accomplish node authentication, an attacker must physically seize a sensor node and collect all of its data. If an attacker obtains a significant number of network nodes, several keys and the secret credential previously held in the network node will be exposed. Consequently, node capture and detection must be resistant to the key management system [99]. As a consequence, each node has its own key and key id, and the base station has an id for each given key. To carry off this attack, the adversary needs to have access to all of the keys and assign a unique set of keys to each of the nodes.

#### 5.3.2 Replay Attack

A replay attack occurs when an active attacker listens in on a legal communication between a sender and a receiver and then transmits the same message to the recipient later in the same session. A timestamp may be included in each message to avoid this.

For communication to be secure, the nonce and a timestamp value must be provided in every transfer and encrypted using the shared key. Once the message has been encrypted, the receiving node checks the timestamp and nonce value to ensure that the message is valid [100]. If the timestamp is incorrect, the message may be rejected by the receiving node.

### 5.3.3 Authentication

Identification of a user's identity is known as authentication in the computing world. An incoming request is associated with a set of identifying credentials, which is the act of connecting those credentials. The authentication procedure begins at the beginning of the program, before the permission and throttle checks are performed, and continues until all other code has been allowed to run. Our approach authenticates the node at several points during the process. Every node is loaded with sensitive information and is ready to be deployed in the field. Authentication is required when setting up a new connection for the first time, and secret information is needed to do this [101]. The primary aim of the protocol is to authenticate nodes and defend them from various types of attacks.

### 5.4 Connectivity

Following the completion of the key configuration, a connected graph is produced. When using the q-composite scheme [41], random, two nodes are deemed connected if they have one common key or a set of q common keys between them. If every pair in the network has a common key, this is referred to as a 100 percent connection. In a probabilistic approach, the probability that every pair in the network shared a key is used to determine the connectedness of the network. Because each pair must share at least q keys in the q composite scheme, it follows that more keys are required to enhance network connectivity. It may enhance network resilience, but it also increases the amount of node and traffic processing required, as well as the amount of bandwidth required for message transmission. When it comes to the E-G scheme, the probability  $P_r$  of sharing at least one key is defined as 1- (Probability, node does not share any key).

$$P_r = 1 - \frac{\binom{P}{k} \binom{P-k}{k}}{\binom{P}{2k}} \quad (5.1)$$

In this scheme, k keys are selected from a pool of P keys. Different values of P, k, and

probability are shown in figure 5.3, as given in [52].  $1-P$  expresses the value of  $Pr$  for  $q$ -composite nodes that share at least  $q$  keys (Node share keys less than  $q$ ). When two nodes share exactly  $i$  keys, the probability is given by the equation.

$$P_{SharedExactly(i)} = \frac{\binom{P}{i} \binom{P-i}{2(k-i)} \binom{2(k-i)}{k-i}}{\binom{P}{k} \binom{P}{k}} \quad (5.2)$$

and  $Pr$  the probability of shared at least  $q$  keys  $Pr$  is given by

$$P_r = 1 - (P(0) + P(1) + \dots + P(q-1)) \quad (5.3)$$

According to our system, the likelihood that every pair of nodes shared a single key is one hundred percent since, in the event that a common key cannot be discovered for any pair, they may create a key using a stored value and stored function in the device.

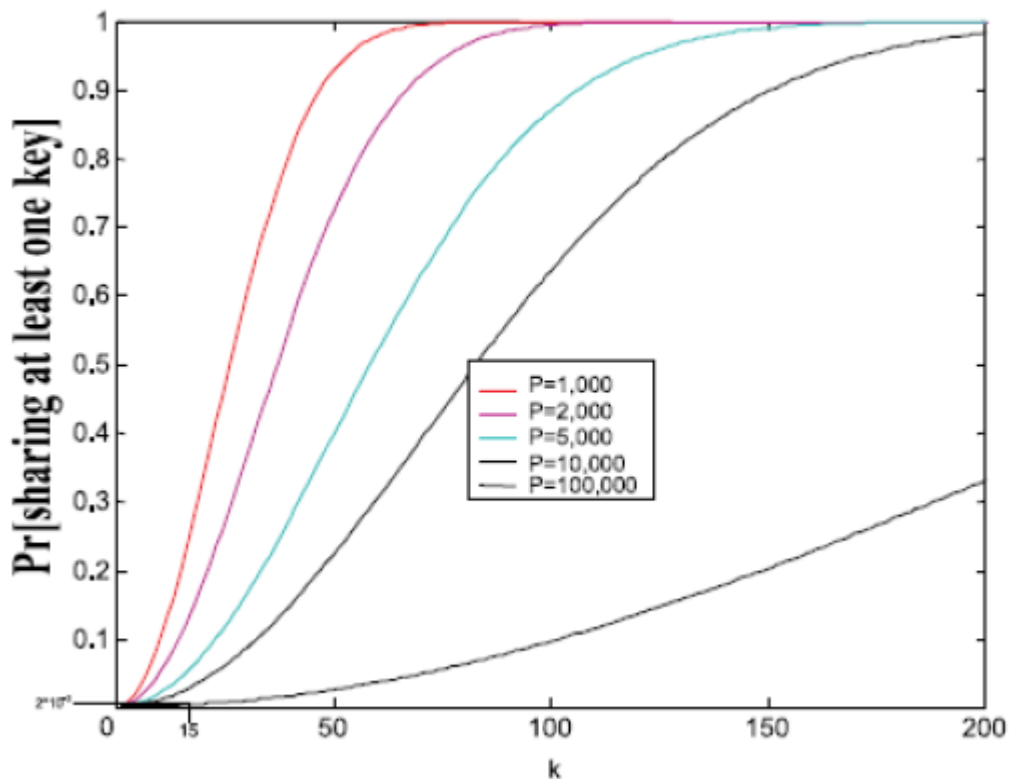


Fig 5.3. Connectivity Graph [52]

### 5.5 Storage Overhead

Sensors usually have a modest amount of memory, around 10KB [102]. To avoid this, maintaining a large number of keys is not advised. However, although storing more keys in a node improves network connectivity, it also increases the chance that a larger number of keys may be compromised if the node is seized. Only 10 to 20 keys are

maintained across scheme nodes, and key hashes are retained when a common key is generated, which has the same impact as removing a key from the scheme. Key hashes are retained when a common key is formed, which has the same effect as deleting a key from the scheme. In addition, each node maintains track of a variable called  $n$  and a hash function, both of which take up little space but are essential for network security.

### 5.6 Resiliency against Node Capture

Node authentication and scheme resilience are two essential requirements of a key management system. When a component or network is hacked, resiliency is defined as a key's likelihood of being disclosed. When two nodes hold the same key from the same chain, according to the study in [103], they may share the key with probability  $(2i-1)/L^2$ . For a given key chain, the probability that  $i$ th key is compromised is  $(k/P)*(i/L)$ . If this chain is compromised, the probability  $P_{ChainComp}$  is given by:

$$P_{ChainComp} = \sum_{i=1}^L \left(\frac{2i-1}{L^2}\right) \left(1 - \left(1 - \frac{k}{P} \frac{i}{L}\right)^x\right) \quad (5.4)$$

A chain's percentage of links that use it is calculated by the ratio of the number of links that use it to the total number of connections created if a chain is compromised. As a consequence, the following factors contribute to the probability of link compromise:

$$P_{LinkComp} = \sum_{i=q}^m (P_{ChainComp})^i \frac{P_{Shared(i)}}{P_{LinkStablish}} \quad (5.5)$$

The chain's length is an important security statistic to consider. Although the calculation may be enhanced, the system's security is proportionate to the length of the chain. To enhance node capture resistance, the chain length should be carefully selected. More processing is required as the key chain is lengthened, but the system's resilience is enhanced. The impact of processing on the key chain is similar to the trade-off between computation and durability, as shown in figure 5.4.

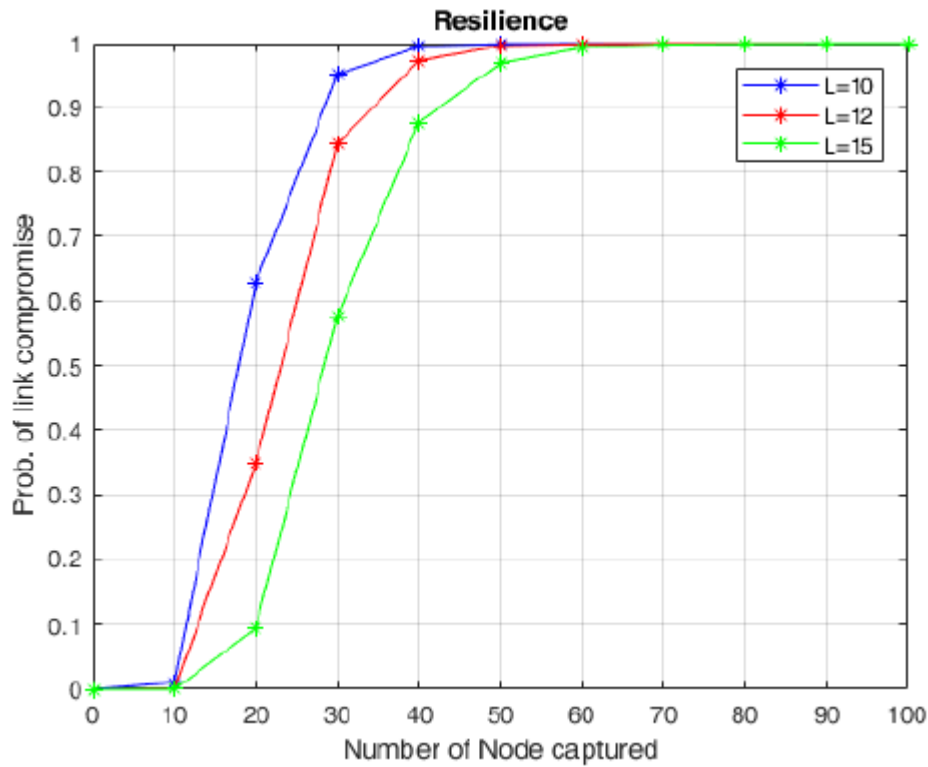


Fig 5.4. Resilience to Node Compromise Attack

### 5.7 Energy Consumption

We implement the scheme in NS-3 and compute the amount of energy it consumes, as shown in figure 5.5. We evaluate the energy consumption of each node or the network as a whole to that of other schemes. We use the following parameters while running the NS3 simulator. The number of nodes in a 300\*300-meter zone ranges from 50 to 500, with a transmission range of 20DB and a key pool of 1000 keys, with 50 keys assigned to each node. The area has a transmission range of 20DB and a transmission range of 300\* 300 meters. Because sensor nodes must be powered by batteries, the key scheme must use as little energy as feasible while fulfilling its intended function. There are a few cases when extra energy is needed if the key is not generated using a recursive approach [104].



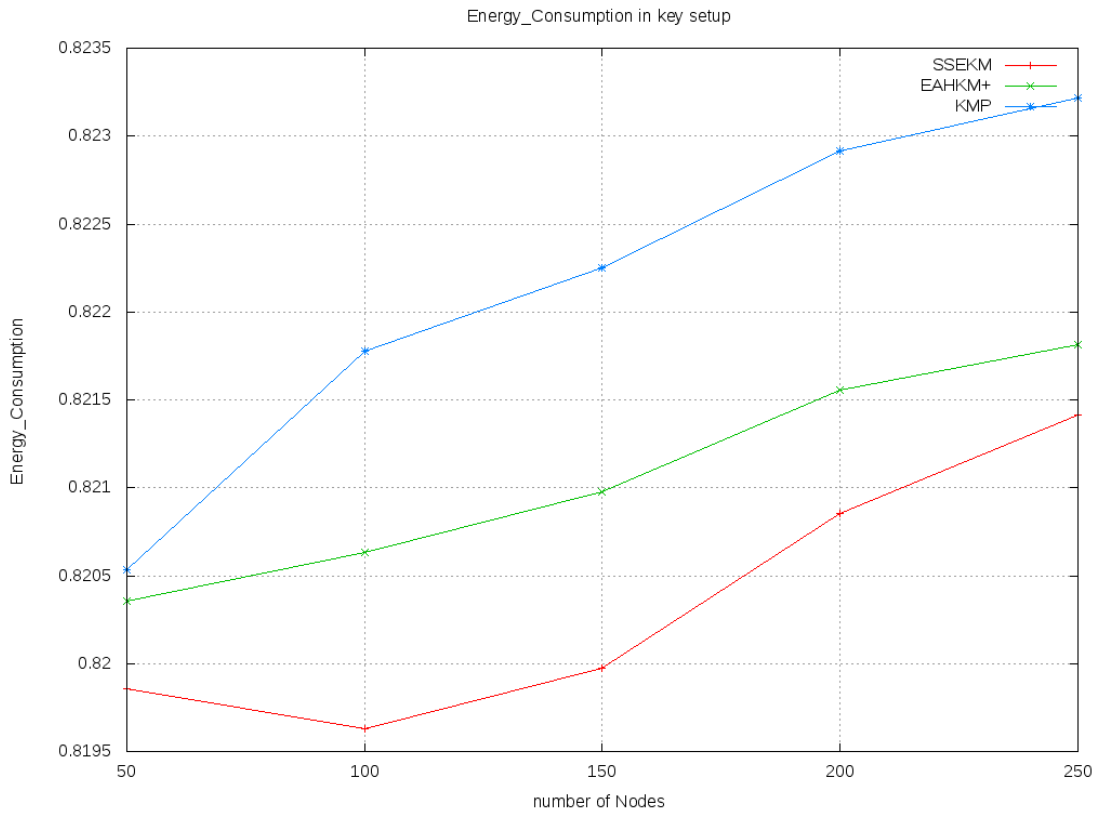


Fig 5.5. Energy Consumption

### 5.8 Summary

The average residual energy in a sensor node and the amount of energy needed to establish the key configuration are utilized to calculate how much power is consumed by this system. Messages from member nodes are often sent to the head over a number of hops, which consumes much more energy than a single hope transfer. Although our approach requires some storage capacity to save the key, in the overwhelming majority of instances, just one message per node is required to establish the key.

## Chapter 6

### Connectivity and Resiliency Analysis

#### 6.1 Connectivity and Resiliency

Sensor nodes are put in the field at random, with no prior knowledge of the location of the next node. Establishing the pairwise key between the nodes and their neighbours is a significant issue in this situation. Two nodes in a sensor networking environment may safely interact with one another as long as they are within range of one another and have a shared secret key. In a big network, keys must be distributed securely. Traditional key techniques such as Diffie Hellman and RSA will fail due to high computational requirements since the node's processing capacity is restricted [105]. A base station that acts as a significant distribution centre sends more data across the network, making it more susceptible to attack at the time of transfer. As a result, pre-distribution, in which keys are pre-loaded into nodes prior to deployment, is the recommended key distribution method. When the nodes are deployed, they must agree on a shared key with the nodes in their surrounding node network. Uploading an excessive number of keys to the node is the thread that, if hacked, may expose a significant amount of information. Furthermore, uploading a small number of keys to the node may result in the network disconnecting. Choosing a suitable key scheme and the amount of keys sent to each node to guarantee connectivity is needed to keep the network connected and less robust.

#### 6.2 Random Graph Model

Dynamic and boundary-crossing computing systems such as WSN do not lend themselves to static key set quality assessments in the same way that traditional computing systems do [106]. This quality metric model, which is also known as a quality metric, may be used in a distributed information system to evaluate key set quality claims, such as reliability. On a broad level, graphs may be used to demonstrate that node A's key set overlaps node B's keyset or that node A's keyset is a valid subset of node B's key set, for example. The graph may then be examined to show, for example, that no intersection of two key sets is fully contained inside the key set of another key set, and so on. When large-scale network graphs with unexpected evolution, such as those observed in the World Wide Web society, are investigated, the situation becomes

more complicated [107].

These are the basic random graph models used to describe objects and relationships as graphs, and they will be covered in detail in this section: The nodes are the representations of the entities. Meanwhile, the relationships between them are mirrored in the boundaries that separate them (e.g., intersection). Many studies have been conducted to determine the function of randomness in the graph model. Many rigorous results have shown that evolving graphs display several interesting and emerging global features. Randomness is more specifically defined as the unpredictability with which the network structure develops daily as a result of the insertion (and deletion) of vast numbers of connections of nodes. Random graphs have the potential to disclose previously unknown features of a network graph because its unpredictability without prior knowledge of possible biases enables the "complete randomness" assumption to be made, which allows for the discovery of previously unknown aspects of the network graph.

In this study, we consider that all connections are symmetric with no direction given. We will not discuss the underlying ideas, but we will give instances of the essential techniques. The number of network nodes is indicated by  $n$  in the following, while the set of all possible edges linking these nodes is denoted by  $n/2$  [108].

- Model  $G(n, m)$ : select the  $m$  edges of  $G$  by selecting them uniformly at random, independently of one another from  $\Omega$ .
- Model  $G(n, p)$ : include each edge of  $\Omega$  in  $G$  independently of the others and with probability  $p$ .
- Model  $G(n, R_0, d)$ : generate  $n$  points in some  $d$ -dimensional metric space uniformly at random and draw an edge between two points only if their distance is at most  $R_0$ .
- Model  $G(k, m, p)$ : each node  $i$  of the  $k$  available creates a set  $S_i$  by selecting uniformly at random each of the available  $m$  objects with probability  $p$ . Then, an edge is formed between two nodes  $i$  and  $j$  only if  $S_i \cap S_j = \emptyset$ . This is the random intersection graph model.

**6.3 Mathematics use in Security:** Prime numbers are very important in the field of network security since they are difficult to guess. Many important cryptographic techniques, such as RSA and Diffie Hellman, are heavily reliant on prime numbers. It is not possible to divide a number by one, for example. It is difficult to come up with a large prime number. The usage of mathematical phenomena such as polynomials, matrices, and recursive formulas in sensor networks is common.

The first polynomial-based system was published in 1992 by Blundo et al. [59]. Using a random number generator, create a polynomial of degree  $t$  over  $GF(q)$  such that  $q$  is a prime integer. Given that the polynomial is symmetric  $f(p, q) = f(q, p)$ , the computed communication keys should be within the range of  $q$ . Every node contains a polynomial part of the function  $f. (i, q)$ .

Dai and colleagues proposed a technique in 2010 [60] that coupled lattice unit decomposition with a polynomial pool-based algorithm. A polynomial pool is created, and an  $L$  matrix is constructed from it. This is an alternate technique for constructing the  $L$  matrix by establishing the key pool over  $GF(q)$ . At the end of the process, the  $U$  matrix is produced on the premise that the product of  $L$  and  $U$  equals the symmetric matrix. The benefit of utilizing a polynomial pool to generate an  $L$  matrix is that mutual authentication may be done during the construction process rather than requiring additional strong authentication.

Matrix-based strategy: It is proposed that scalability limitations be solved using matrix-based techniques at the cost of computing time. Matrix decomposition, such as LU matrix decomposition, is utilized in many matrix-based techniques; the matrix most often employed in this scheme is symmetrical in nature. Blom's technique (Blom, 1985) [54] is an early 1980s symmetric key generation method (SKGS). Every node in a network may interact with every other node in the network, even if they have less information. This is referred to as a threshold technique of secret disclosure. As previously stated, the secret has been divided and dispersed across the network's nodes. A network is exposed to the attacker when just a limited number of nodes are seized or when a certain threshold is achieved. The issue with this method is that when the number of sensor nodes gathered exceeds certain criteria, the entire system is exposed. Blom's Scheme is only known to the KDC, and it employs two matrices over  $GF(q)$ : a public  $(\lambda+1)n$  matrix  $M$  and a secret  $(\lambda+1)n$  symmetric random matrix  $D$ , both of which

are known only to the KDC. These matrices must be used to construct the symmetric  $n \times n$  matrix  $K = (DM)^T M$ , whose element  $K_{i,j} = K_{j,i}$  corresponds to the key linking the graph's nodes.

Eltoweissy et al. [56] developed the Exclusion Basis Systems EBS, which used combinatorial optimization for key management in its development. EBS is very scalable when used in big networks. EBS is a game in which the outcome is determined by two variables,  $k$ , and  $m$ , which are interconnected. For an EBS configuration to be able to handle a set of  $N$  nodes, it has to contain a set of  $k+m$  keys. Each node knows a total of  $k+m$  keys out of a total of  $k+m$  keys in a total of  $k+m$  keys. Nobody should be aware of the same set of  $k$  keys as anybody else. Any new node may be added as long as the node that is being replaced maintains access to a separate set of  $k$  keys; otherwise, no new nodes will be created. Using new keys distributed with  $m$  keys that the hacked node is ignorant of, it is possible to expel the node from the network. In order to suit the needs of the network, the values of  $k$  and  $m$  may be changed as needed. In order to evict a hacked node, new keys are disseminated by disseminating old ones that the compromised node is not familiar with. Clearly, as the value of ' $m$ ' grows, so does the amount of time spend conversing? Collusion attacks (A type of attack) are very dangerous to the EBS system. A collusion attack is an attack where the malicious user (i.e., untrusted user) gives the wrong or misleading feedback about the cloud service provided by the cloud provider. Younis and colleagues [109] have created a method that has the ability to protect EBS-based key management systems against collusion assaults. A representation of EBS key distribution is given in table 6.1.

Table-6.1 EBS Matrix Example

	N0	N1	N2	N3	N4	N5
K1	1	1	1	0	0	0
K2	1	1	0	0	0	1
K3	0	0	1	1	1	0
K4	0	0	0	1	1	1

#### 6.4 Resiliency and connectivity in Random Key distribution

When assessing the security of sensor networks, it is critical to consider their resilience. Suppose a section of the keys used in the network is revealed, then how big of an effect

this will have on the remaining portions of the network. Any compromised network key does not affect the other keys in the system, which is a strong, resilient system guarantee. In networks, scalability is a problem because as the network expands, we must use more keys, which increases the complexity of the system. When many keys are used, connectivity is improved, but resilience is reduced, and vice versa.

Random key distribution involves selecting keys from a huge pool and storing them in the node, following which nodes are deployed at random intervals in the target field. When utilizing the random graph function method, placing too many keys onto a node may improve resistance and create a thread in the system since a compromise of a node would expose a negligible fraction of all keys. We may accomplish the same degree of security by loading just a limited number of keys into the node, improving the node's resilience. Because nodes are deployed at random, there is no need to exchange the key with each node. As a result, if all nodes are spread in various regions and groups, a tip about loading the key from a certain range of nodes may be feasible. This pool is archived by dividing it into  $N$  overlapping sections, each of which is archived individually.  $N$  may have a value ranging from 1 to  $x$ . Keys are assigned to various pools at random. Arrange the groups in a random sequence such that the adjustment node with the fewest keys includes the most often encountered keys.

$G = (V(G), E(G))$ , where  $V(G)$  indicates a (nonempty) set of vertices and  $E(G)$  denotes a set of edges, i.e., a set of ordered pairs of vertices. For the sake of this study, all graphs are considered to be undirected, simple, and finite; for a more detailed introduction to graph theory, see [21]. A random graph  $G$  on the vertex set  $V_n = 1, \dots, n$  is a graph-valued random variable (rv) defined by  $G(n): G(V_n)$ , where  $G(V_n)$  is the set of all simple undirected graphs on  $V_n$ . Given a probability triple  $(G, F, P)$ , a random graph  $G$  on the vertex set  $V_n = 1, \dots, n$  is a graph-valued random variable (rv) defined by  $G(n): G(V_n)$ . For pure random distribution scheme like (EG scheme) we have to allocate a big key ring to every node for adequate connectivity level. The relationship is given by

$$P_c = \lim_{n \rightarrow \infty} \Pr[G(n, p) \text{ is connected}] = e^{e^{-c}} \quad (6.1)$$

Where

$$p = \frac{\ln(n)}{n} + \frac{c}{n} \quad \text{and } c \text{ is any real constant}$$

In our model, every node has a position knowledge probability  $P_{loc}$  between 1 and 0. If the node knows the position exactly, then the value is one; if none has no knowledge about the position  $P_{loc}$  is 0. Instead of making a big pool of key, we make N pools with a smaller number of keys in every pool. Nodes are given the keys according to the group number. From a big pool P, if key ring k is selected, the probability is given by

$$p' = 1 - \frac{(P - k)!^2}{P!(P - 2k)!} \quad (6.2)$$

Instead of creating a single pool, n pool will be created n pool P1, P2..Pn. The different numbers of keys are selected from different pools. There is a special selection of keys for overlapping areas. These keys are selected from two adjacent pools, like P1+P2 and P2+P3, as shown in table-6, and deployment is shown in figure 6.1.

Table 6.2 Key Distribution for Overlapping Area

Key Distribution		For overlapping area	
Pool	Key Select	Pool	Key Select
P1	k1-x	P1+P2	2x
P2	k2-x	P2+P3	2x
P3	k3-x	P3+P4	2x
...	...	...	...
Pn	kn-x	Pn-1+Pn	2x

In this case, the common key probability is given by

$$p' = 1 - \frac{(P1 - k1 - x)!^2}{P!(P - 2(k1 - x))!} \quad (6.3)$$

Different pools use different keys from pools. Because keys are coming from a small pool, it gives better connectivity.

### 6.5 Analysis of Connectivity and Resiliency

When compared to the conventional EG method, assigning keys to a small pool leads to better connection and better resilience. This technique is also scalable since just a few keys are required for a large network to operate correctly. The resilience and connectivity of key management schemes, as well as their overall efficacy, are assessed. In the pairwise connection model, the number of groups and the number of keys in each

group are significant variables. Another similar analysis is given by Sarkar et al. [110] for clustered networks.

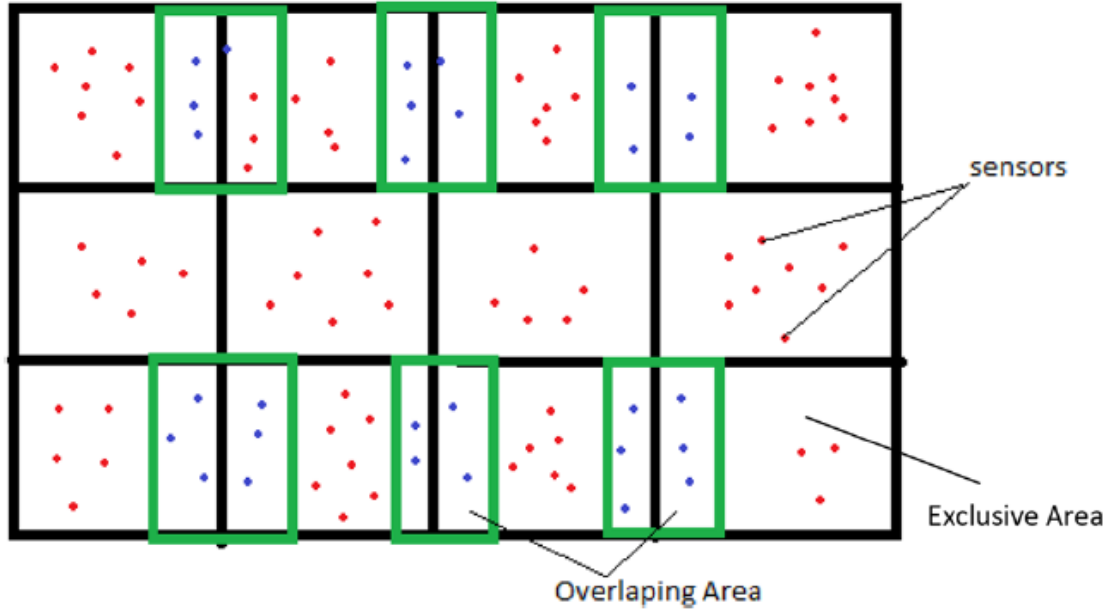


Fig-6.1 Node Deployment

**Connectivity:** By using random graph theory, we can analyze the coverage and connectivity of random deployment. Bernoulli random graphs  $G(n, R)$  use graph theory. A key result is pertaining to  $G(n, R)$  geometric random graph. In normal network connectivity, ensure that every pair of nodes have a path, as shown in figure 6.2.  $K$ -connectivity ensures whether there exists  $K$  disjoint path between the nodes. For a secure connection, there must be a share of common keys between the nodes. Connectivity also depends on the transmission range. According to research for the high probability of connectivity, the critical transmission range is

$$o \sqrt{\left(\frac{\log n}{n}\right) \cdot \log 1/4 n} \quad (6.4)$$

For any sensor network connectivity is given by

$$c = \frac{\text{number of communication links present in the network}}{\text{total number of link in the network}}$$



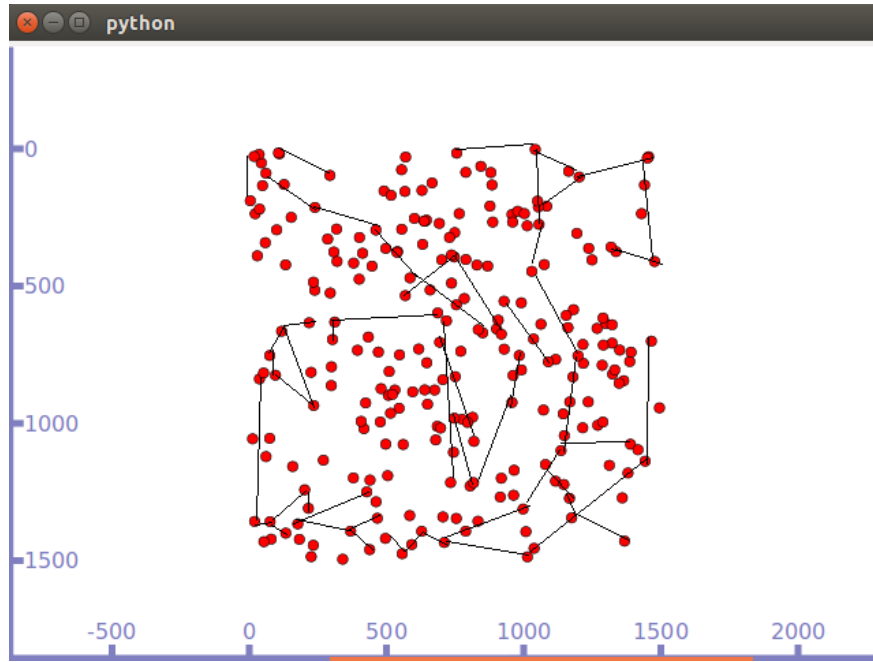


Fig 6.2: Connectivity of Random Graph Model

In probabilistic systems, it is determined by the number of pairs of keys that are shared in order to establish a safe connection; therefore, it is determined by

$$p' = 1 - \frac{(P-1-k)^2}{P(P-2k)} \quad (6.5)$$

**Resiliency:** It is critical for the security of sensor networks that nodes be identified and authenticated. The bulk of attacks may be avoided by ensuring that each and every node is properly authorized and protected. An attack on a node's control is one of the most severe threats to the Internet's stability (WSN). Following the seizure of specific nodes, any previously hidden keys contained inside them are made visible to the user, and any connections that were previously connected to these keys are severed as a consequence. As a result, shown in figure 6.3, the nodes become more separated from one another, and the network as a whole suffers as a consequence. When a network is subjected to a node capture attack, the robustness of the network is what determines its level of resistance. In order to assess the resilience of a network, it is necessary to consider two parameters: the portion of the network that has been hacked and the likelihood that more parts of the network may be compromised over time. For random deployment, if a node is compromised  $k$  keys are revealed, and a portion of  $k/P$  is compromised [52]. The probability that a link is not compromised is given by  $1-k/P$ . And in the case of the

t node, compromise probability is given in equation 6.6.

$$P_{com} = 1 - \left(1 - \frac{k}{P}\right)^t \quad (6.6)$$

In the proposed scheme,  $k-x$  nodes are selected from pool  $P$ , and  $2x$  nodes are selected from  $2P$  for the overlapping area. Therefore, probability is given by  $1-x/P$  or  $1 - (1-(k-x))/p$ . The probability of key compromise is given in equation 6.7.

$$P_{com} = 1 - \left(1 - \frac{k-x}{P}\right)^{\frac{t}{2}} \left(1 - \frac{x}{P}\right)^{\frac{t}{2}} \quad (6.7)$$

This distribution gives better resiliency with the same key ring.

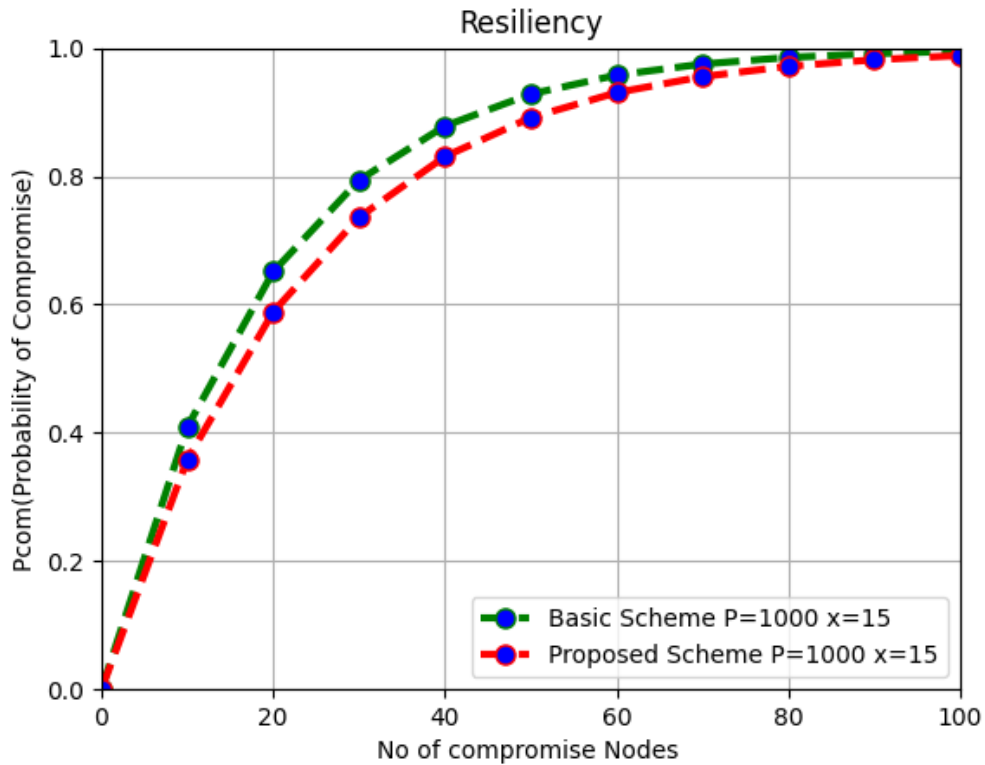


Fig 6.3: Resiliency of Proposed Model

**Authentication and verification:** It is possible to authenticate from one node to another using a paired method, which allows sensors to identify the neighbours with whom they are talking. When nodes authenticate with one another, it may be easier to identify rogue nodes and improve resilience to node replication attacks [111], which has a big impact on network security. Furthermore, since each communication link has its own set of keys, an attacker will only compromise the incident's communication connections to

the seized sensors if they manage to grab a group of sensors; this is known as the "perfect resilience" feature.

**Scalability:** Key management is considered to be scalable if it can accommodate the addition of more nodes without negatively impacting the existing network. According to our implementation, Connection KPS is not scalable since the system will not function properly if the number of new nodes to be deployed is not known in advance of their deployment. When the network's connectivity key for connection and shared connectivity key set is selected from a key pool by a suitable previous selection of the network's characteristics before the new nodes are included, more nodes may be accommodated in the network. We, on the other hand, have developed a scalable communication method that does not require any prior knowledge of deployment techniques.

## **6.6 Summary**

This chapter provides the methods to analyse the connectivity and resiliency of the key management method in ad hoc and sensor network environments. Many mathematical methods are discussed throughout the chapter and how they might be applied to this proposed scheme. With this study, the primary goal was to develop efficient key management systems, a method to enhance resiliency so that limited loss is there when a certain portion of the network compromise. To achieve the objectives, some methods make use of random systems that have been deliberately created (either deterministically or probabilistically). The field of wireless network security has progressed to a stage at which theoretical foundations can be applied to formulate design principles. This maturity is based on the breadth and depth of results found in the reviewed papers. The investigation into one feature of sensors networks that differentiates them from traditional networks, on the other hand, has not yet been completed in its whole. This refers to a lack of structure and how it impacts how assaults propagate throughout a network and the attempts to restore order. It is essential to find a solution to the key management issue, one of the difficulties. However, since there is no structure in place, it is still unclear how to deal with large-scale assaults on mobile networks, which are becoming more common. As a result of the absence of network information accessible, it is not feasible to determine the locations of attacks or the patterns of attack distribution. Conventional networks, where various models of

virus/disease transmission have been developed to solve the issue, are more susceptible to this problem. Because of the absence of structure in mobile networks, it is necessary to re-examine all models and methods designed for use in these networks.

The expressive and deductive capabilities of various formalisms for expressing trust/security-related features of large combinatorial structures were also investigated in this chapter, emphasizing attributes associated with significant groups of network nodes being the focus of the investigation. Keyset characteristics are a limited number of predicates that, under specific circumstances, restrict the limiting behavior of systems. After then, the features may be codified with the assistance of first- and second-order graph languages, depending on their complexity. Extension statements may specify the circumstances under which the model displays 0–1 behavior as long as their characteristics can be stated in first-order graph language. Therefore, all features are asymptotically true with a probability of 0 or 1 for a variable. In their respective instances, all of the characteristics are asymptotically true. A second-order logic fragment 0–1 behavior vocabulary may explain characteristics that cannot be stated in the first-order graph language.

## **Chapter 7**

### **Key Management for Hierarchical Sensor Networks**

#### **7.1 Introduction**

Hierarchical networks make the hierarchy with the help of cluster heads (CH) and base stations. In these networks, we can take advantage of higher resources of CHs. Cluster heads (CHs) are the highest-ranking hierarchy members with the most storage and processing power. Sensor nodes, and from the other side, are at the bottom of the hierarchy and have fundamental capabilities like energy and storage at first. Based on their capabilities, hierarchical wireless sensor networks (WSNs) are divided into two kinds. Because it allows different applications to utilize the whole or a portion of a wireless sensor network in a cluster-based hierarchical structure, this architecture is suggested for usage in wireless sensor networks. Each application's network, processor, and protocol needs may be distinct. This network is split into groups, each of which is structured hierarchically. When various risks and attack scenarios are examined, the Wireless Sensor Network (WSN) will be investigated more deeply [112]. Due to the restricted resources available to sensor networks, attackers are increasingly physically compromising sensor nodes, requiring establishing an environment that employs effective key management methods. WSN will certainly assess the effectiveness of symmetric key management methods. Traditional key management techniques such as asymmetric key cryptosystems and Key Distribution Centres (KDC) are difficult to set up and maintain in a network context. In a hierarchical system, a cluster head may serve as the primary distribution point for the whole system and works as KDC. In these situations, data security and authentication are essential. The success of wireless sensor networks depends on a well-functioning key management system [113].

#### **7.2 Network Model**

The term "clustered network" refers to a network in which groups have been established and where a cluster member's CH is in charge of each group. The distribution of keys hierarchically takes place. The session key is transmitted securely via the use of a master key. The nonce is used to prevent duplicates from occurring. At the base station [114], there is a key distribution center (KDC) where keys are distributed. KDC is divided into two layers in terms of functioning: a cluster head and a global controller. The model nodes in this network are grouped inside a building or a region of the

network. Each node is given a cluster leader, who also serves as a local key distribution centre (KDC). There are many nodes with a lot of processing capability in this network [115].

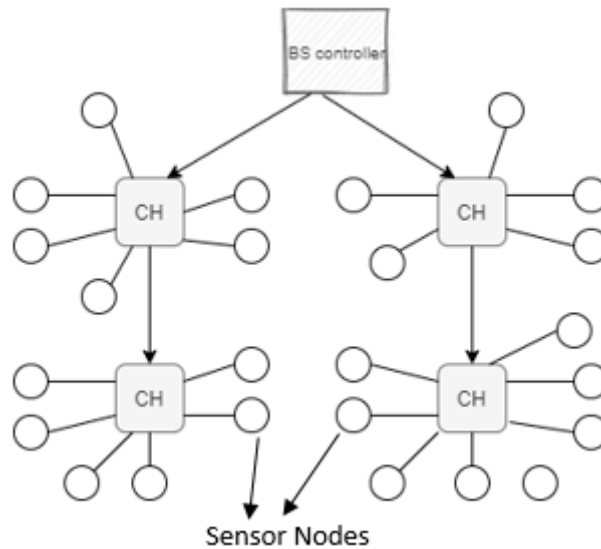


Fig 7.1 Network Model

A unique id identifies each node. The creation of master keys consists of the following steps: A function and an initial key are stored in the node. Node makes use of the function and the first term to generate the master key. When a node needs a session key, it requests the localized or global key distribution centre (KDC). Every time Node, one communicates with another node. It requests a session key for the KDC from that node. Allowing for the possibility of communication across nodes in a single cluster, consider the following scenario.

### 7.3 Proposed Scheme

The term for this kind of communication is intra-cluster communication. The cluster chosen as the KDC is a session key generated following the technique described in figure 7.1. It is also shown in the following algorithm.

#### **Intracultural Communication:**

A node sends the request to the local KDC by sending his id and id of the node whose it wants to send a message.  $IDA \parallel ID B \parallel N1$ . Local cluster reply according to the algorithm is as follows.

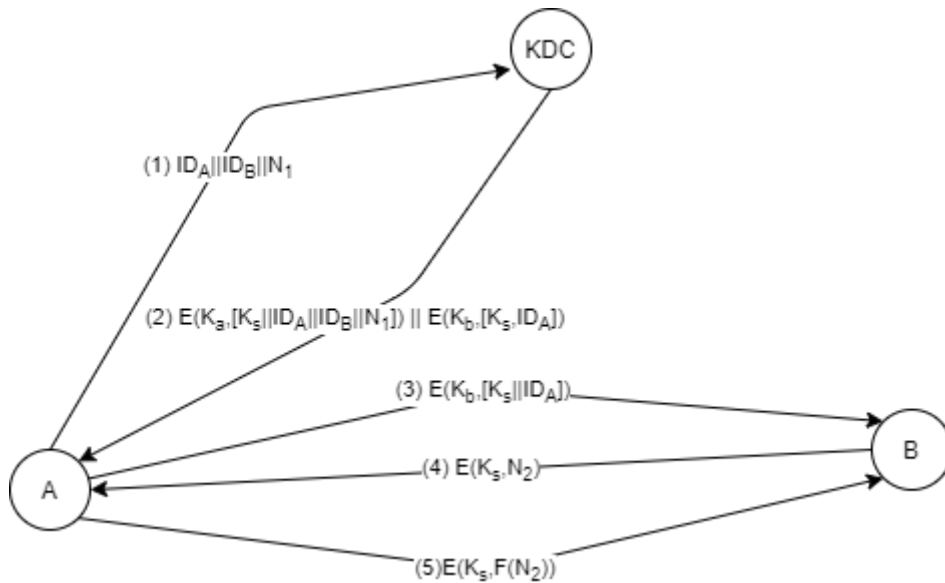


Fig 7.2. Session Key Generation

#### Key Generation Algorithm

1. Before deployment, every node has a function and initial term. That is updated in every transaction.
2. When node A wants a session key to communicate to node B, it sends a request to KDC for the session key by sending the id of A and B and a nonce.
3. KDC sends the session key encrypted by the master key of A. It also includes an encrypted id of A to send to B.
4. A sends an encrypted message to B so that B can also get the session key and id of A.
5. B sends a nonce encrypted by the session key.
6. A send nonce and conform by acknowledgment.

For inter-cluster communication, the global cluster head chose KDC.

#### New Node Addition:

When a new node joins the network, it must first get credentials from the base station or the controller. When the message reaches any cluster head, it may be verified by forwarding it to the BS server. The cluster head allows the node to join the system once it has been confirmed by BS. When a cluster node needs to communicate with another cluster node, the cluster head is utilized as the KDC, and when there is intercultural

communication, BS is used as the KDC. Figure 7.2 shows the exchange of information between communicating entities.

#### 7.4 Observation and Analysis

We analyze and compare the scheme on the bases of these parameters' connectivity, resiliency, storage, and computation. Authentication of A and B did by KDC. The nonce is used for non-reproduction. A simulation study is done on NS3 and energy consumption and network lifetime. The result of energy consumption is shown in figure 7.3.

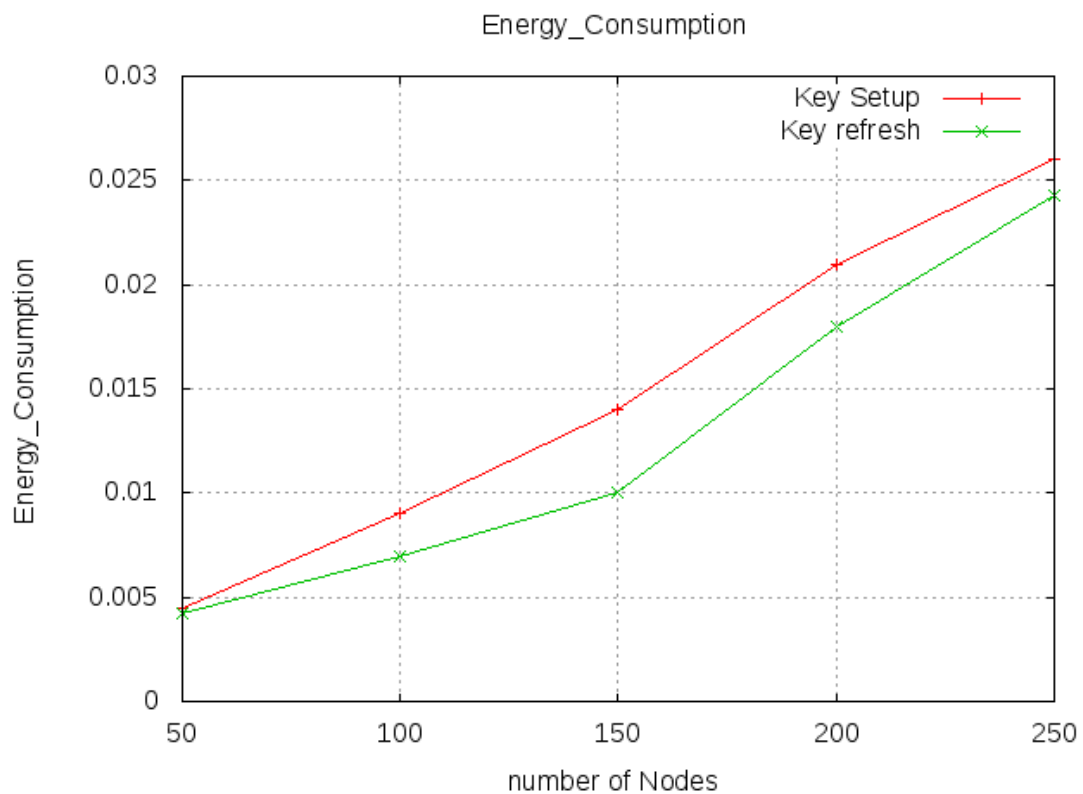


Fig 7.3. Energy Consumption in Key Setup

#### 7.5 Security Analysis:

For any security scheme, along with resource requirements, it is also important that the scheme must fulfil the security parameters and prevent various attacks. We analyse the scheme for various types of attacks.



### **Masquerading attack**

One of the most important security criteria of the key exchange protocol is the ability of both parties to authenticate one another. This is one of the most difficult needs to meet. If node A initiates the communication with node B, how can A be confident that the response received by A from B is really from B and not from someone else masquerading as B? B must be convinced that the message is coming from A well as the other way around. The cluster leader is responsible for gaining the trust of A and B in the group. When it comes to cluster heads, the only one who can be recognized knows the master key for both A and B. Only A has the ability to see a message encrypted with a secure key sent from the cluster head, and only B can read a message encrypted with B's secure key received from the cluster head. In step 3, an opponent has the option of saving a packet and replaying it later; nevertheless, the nonce differentiates between the original and new packets sent. The scheme's security is jeopardized due to vulnerabilities in the cluster head and the master key.

### **Replay attack:**

An attacker may intercept the packet and subsequently transmit it as though it came from the authorized party. To prevent replay attacks, the nonce may distinguish between old and new requests. If an opponent captures the packet in step 2 and subsequently delivers it to A, A may recognize it as an old request and reject it. A packet in step 3 lacks a nonce and may therefore be intercepted and sent to B by the adversary. In step 3, a timestamp may be appended to the packet.

### **Eavesdropping:**

The inability to reload messages is because all chats are encrypted using a secret key. While the IDs of A and B do not cause any issues in step 1, the communication may be encrypted in step 2 using a secure key shared by the cluster chiefs of A and B. The first message sent from A to be, like the previous message, is encrypted using B's secret key, and the content of the message cannot be disclosed. Only the finished packet may be recorded to avoid a replay attack, which is avoided by giving a timestamp.

## **7.6 Simulation Results**

According to our tests, the system is immune to a variety of assaults, including the replay attack and the node captures attack. In addition, simulation research was carried out to determine the energy used by the key setup and refresh. It is more resistant to

node capture attempts, according to the scheme's security study. To test the scheme in NS3 simulator, we use the following parameters: sensors nodes are deployed in a 300\*300-meter area, with a transmission range Tx Range of 20DB and a key pool of 1000 keys, 50 of which are assigned to each node. We keep track of the amount of energy used. The energy consumption in this system is calculated using the average residual energy in a sensor node and the energy needed for the key setup. Messages transferred from a member node to the head are often sent across several hops, consuming more energy than a single hope transfer. Although some storage is needed in our system to keep the key, most of the time, just one message per node is required to configure the key. If the key is not generated via a recursive method, more energy is needed in certain circumstances. In terms of results, this technique beats current approaches.

## **7.7 Summery**

Meeting high-security standards with little resources is one of the challenges of WSNs. Node authentication, data confidentiality, anti-compromise, and traffic analysis resistance are all requirements for WSN security. Sensor network security is crucial and must be done correctly. One system has confidence in another device and between computer and human in an ever-expanding world of linked devices. To make the system trustworthy, it is essential to fully comprehend its security issues and carefully implement the security technique. Many open securities-related cases require in-depth investigation. The real success of sensor network technology is measured by either eliminating a poor situation or maintaining a good one. Message secrecy and authentication are two significant concerns that must be solved. For hierarchical sensor networks, we can take advantage of high-energy nodes by making them cluster heads and giving them more responsibility. At the same time, cluster heads require more security as they can be a single point of failure for all members' nodes of the clusters. A heterogeneous wireless sensor network's security approach may be adapted to IoT and vehicle networks [116]. Sensors, connected devices, and control devices are all part of an IoT network. Security is essential for modern applications and new types of networks [117]. In these networks, authenticating the device and securely transmitting data is a major concern.

## **Chapter 8**

### **Dynamic Key Management for Modern Applications**

#### **8.1 Introduction**

The Internet is the backbone of the modern world. It is present everywhere, controls almost everything, and gives us benefits of all sorts that just were not possible before. It now connects everything with everything – people and devices included. In reality, IoT is a simple idea that involves connecting everything on the planet to the Internet through wireless networks. The IoT refers to a large group of linked items and people who gather and exchange data about their use and the area they are located. Surprisingly, there are a staggering amount of these connected gadgets [118]. From intelligent microwaves that automatically cook your own food to self-driving cars with sophisticated sensors which identify obstacles. Then, recommend ideal training plans for smart refrigerators to store your food most efficiently.

Man has lived on this planet for a long time, and every time he tries to make life safer and easier by using different methods. Human beings are encouraged to observe things and do experiments and see how things work, and by using the results of these experiments, he adopts the changes in the lifestyle for a safe and easy life for the next generation. One of the methods for making things better is monitoring the process and collecting the results, and also taking the feedback and improving the system [119]. In modern life, many methods and techniques are used to the monitoring of activity, behavioural patterns, or any other change in environmental conditions. These methods include different sensors, audio-video devices, and surveillance from a distance by means of electronic equipment (such as CCTV cameras) [120]. IoT changes the way how we use using the internet for taking information, and now it makes things automatic and less human interaction. Smart surveillance includes not only collecting the data but also analysing the data and making some decisions. It also includes detecting abnormal behaviour or pattern and triggering the alarm. Companies and organizations push for greater transparency across IoT security and data policies [3].

#### **8.2. Heterogeneous and IoT Networks**

It is possible to link computers and other devices running multiple operating systems or using different communication protocols in a heterogeneous network. Local area networks (LANs), for example, are diverse in that they link computers running

Microsoft Windows and Linux operating systems with machines running Apple Macintosh operating systems, among other differences. A clustered network comprises similar devices, each of which has a cluster head with increased capabilities and processing capacity. An IoT network is an interconnection of different types of things capable of receiving signals and receiving command from a human or machine [121]. There are significant differences between WSN and IoT. Whereas the main problem of WSN is limited computation power and limited battery, the primary concern of IoT is heterogeneity. In the IoT network, different types of devices create a problem in interoperability and connectivity [122].

### **8.3. Heterogeneous and IoT Networks Applications**

With the advent of ultra-low-cost electronic chips and the widespread availability of computer networks, everything from a tablet to an aircraft may become a member of the Internet of Things [123]. The following are a few examples of such applications:

**8.3.1. IoT at Homes:** Smart homes are the most important and efficient application to consider when it comes to the Internet of Things. It is the most widely used IoT application in virtually every form of communication [124]. The Internet of Things enables the notion of home automation, which may encompass, among other things, lighting, heating and air, media, and security systems. It may also provide long-term advantages, such as electricity costs, by shutting off equipment when not in use. Actuators and sensors will aid in the efficient and effective use of energy while also offering extra comfort in everyday life. These sensors will keep track of the outside temperature and the number of people in the space, enabling them to control the amount of heating, cooling, and lighting supplied, among other things. This strategy would result in cost savings as well as increased energy efficiency.

**8.3.2. Wearables:** Users of Internet of Things apps are still interested in smart homes, which are akin to cities. Worldwide, consumers look forward to Apple's newest smartwatches every year. Another example is the Sony Smart B Trainer, LookSee bracelet, and Myo gesture control. These wearable devices make our life simpler and more convenient. To give an idea, some examples are smartwatches, pacemakers, and googles.

**8.3.3. Smart Cities:** Large cities provide a higher demand for energy, water, buildings, public areas, transportation, and other resources. As a result, we must seek "smart"

solutions, which must be successful and feasible. The Internet of Things (IoT) plays a critical role in improving city smartness. [125].

**8.3.4. Smart Grid:** "Smart grid" refers to an electrical grid that is designed to increase power transmission efficiency and improve the economics and reliability of electricity distribution. Many pieces of energy-consuming equipment already have an Internet connection, which allows them to communicate with utilities in order to balance power output and energy consumption. The devices may also be connected to sensors that provide data on power use to a central server on a regular basis.

**8.3.5. Smart Vehicles:** With the increase in the standard of living of people, vehicles and traffic on roads has increased, and so has increased the number of traffic accidents. One way to minimize accidents is by the use of M2M communication and smart cars. McGill University has developed a pilot program to test a remote control vehicle in order to reduce the probability of a car accident and a human error. These self-driving cars will provide more than just safety since they will save time and reduce the stress associated with driving, among other benefits. According to the Institute of Electrical and Electronics Engineers (IEEE), autonomous vehicles will account for up to 75% of all vehicles on the road worldwide by 2040.

**8.3.6. Health Care:** There is a variety of Internet of Things (IoT) applications in healthcare, ranging from remote monitoring equipment to advanced and intelligent sensors to equipment integration. Other advantages offered by the Internet of Things technology to the healthcare sector include the monitoring of items, personnel, and patients, identification and authentication of individuals, automated data gathering, and sensing, among others [126]. The study of different bodily characteristics via the establishment of a wireless body area network is an important subject that makes extensive use of the Internet of Things. A Body Area Network (BAN) or a Wireless Body Area Network (WBAN) is simply a wireless network of wearable electronic devices or gadgets, biomedical sensors, and wireless communication devices. These short-range wireless networked devices can be positioned in, on, and around the body and are used to collect and/or monitor vital parameters of the body, which are then transmitted outside to a WLAN, or Internet or to a united central database where all information collected is processed.

**8.3.7. Securities and Emergencies:** Other uses for the Internet of Things include security and surveillance, such as space surveillance, monitoring of people and assets, infrastructure and equipment maintenance, alerting, and so on. Using Internet of Things (IoT) technology in security and emergency response has increased significantly in recent years, including perimeter access control, liquid presence detection, radiation levels, explosive and hazardous gas detection, and so on. Peripheral control is often used to monitor and prohibit unauthorized access to restricted areas inside the perimeter of a structure. Liquid presence monitoring keeps data centres, warehouses, and sensitive building grounds operating smoothly and free of corrosion. While a nuclear power plant is in operation, the radioactivity monitoring application monitors radiation levels in the surrounding area to give leakage warnings. Finally, the Internet of Things application monitors gas levels and leakage in industrial settings, chemical manufacturing environments, and mines. Problems with Internet of Things-based Wireless Body Area Networks are given in [127].

#### **8.4. Security requirements of IoT Networks**

For any network system, basic security requirements are authentication, confidentiality, and integrity [128]. It is required in the network that devices can authenticate each other and data transmission between them is secure without any modification. Most of the attacks can be prevented by using authentication methods and encryption. Different authentication mechanisms like a private key, public key, and MAC code or HASH code required keys. Other security requirements are access control and non-repudiation [129]. Following are the security requirements in IoT networks:

**8.4.1. Privacy and Security:** Any technological development in the modern day is fraught with anxiety and problems regarding privacy and security concerns. When a network has many connected devices, the importance of network security increases; even more, several ways of attacking the system are available, including restricting network availability, introducing incorrect information into the network, and obtaining personal data. It is challenging to enforce adequate security and privacy measures using the techniques available at this time. Furthermore, since the Internet of Things uses various item identification techniques, such as RFID and 2D barcodes, it is critical to incorporate adequate privacy protections and prevent unauthorized access.

**8.4.2. Data Storage and Intelligence:** Cleaning, analysing, and interpreting the enormous quantities of data gathered by the sensors is another challenge in creating Internet of Things applications. To develop smart IoT applications, the data gathered by IoT devices should always be maintained and utilized correctly. Artificial intelligence algorithms and machine learning methods derived from social programs, genetic programming, artificial neural, and other artificial intelligence approaches are required to accomplish automation. Wireless Sensor Networks are being investigated as a method of assessing data. [130]. Such systems exchange data between sensor nodes, which is then sent to a decentralized system that the collected sensory information can measure.

**8.4.3. Quality of Service (QoS):** The two most significant variables affecting the service quality (QoS) of IoT systems are throughput and bandwidth. The Internet of Things (IoT) generates a wide variety of data, from sensors connected to machine components or environmental monitors to the phrases we scream at our smart speakers in large numbers. To send data over the wireless channel, devices will need to utilize a certain frequency. Establishing quality-of-service guarantees in wireless sensor networks is challenging due to limitations in allocation and abilities to make in shared wireless media. [131]. Another major research subject in cloud computing is quality of service, which will become more essential as the data and tools required for the Internet of Things become more readily available on clouds.

**8.4.4. Interoperability and Standardization:** The wide range of IoT devices, both in hardware and software, makes creating apps that operate consistently across disparate technological ecosystems challenging. Because different manufacturers will create devices, technology and services designed for one device may not be available to other devices in the future [132]. Consequently, Internet of Things standardization is essential to better interoperability for all networked items and sensor devices.

**8.4.5. Object's safety and security:** Because the Internet of Things comprises a huge number of perceptual items that are dispersed over a wide geographic region, it is essential to prevent unauthorized access to the objects that may cause physical harm to them or cause their function to be altered [133]. Wireless sensor networks have a number of security needs that are

Table 8.1 Security Requirements for IoT

S.N.	Security Parameter	Description
1	Authenticity	Guarantee that communicating nodes can verify each other
2	Confidentiality	Transmitted data not seen by attackers
3	Integrity	Transmitted data not altered
4	Scalability	Security schemes should be applicable to large networks
5	Flexibility	Allow to add or remove the nodes from deployed networks

### 8.5. Security Challenges

Highly constrained devices that have limited resources use low bandwidth slanted and wireless channels. Data flows freely in these open channels, so it is very important to implement security properly in these systems. These are the main challenges for these systems:

#### Limited Resources and Constrained Devices

Various IoT gadgets have limited storage capacity, memory, and handling ability and typically need the option to work on declining vitality when strolling on batteries, for example. Insurance methodologies that rely heavily on authentication are not an excellent match for these restricted tools due to the fact that they are not currently designed to perform complex encryption and rapidly adequate decoding, which enables you to transfer records securely and gradually. Such tools are constantly at risk for aspect channel assaults systematic power analysis assaults, which can be used to reverse such calculations. On the other hand, low-weight encryption calculations are normally less difficult to lease controlled gadgets [134]. IoT implementations must use two or three protective layers, such as isolating gadgets on separate systems and using firewalls, to penalize these gadget constraints [135].

#### Vulnerabilities Detection and Incidents

Security vulnerabilities and ruptures are unavoidable regardless of the best efforts. How would you know if it has undermined your IoT framework? Large-scale Internet of Things systems, particularly those with a diverse range of connected devices and a diverse range of devices, applications, administrations, and communication protocols, may make it difficult to determine when an incident has occurred due to the system's multifarious nature, both in terms of the number of connected devices and the diversity



of devices, applications, administrations, and communication protocols involved. Systems for detecting vulnerabilities and breaks include monitoring system variable changes and activity log data abnormalities, using protection viewpoint and inquiry to notice and notify when episodes occur and engaging in infiltration testing to reveal weaknesses. Additional mechanisms have been put in place to cope with various risks and failures that may occur [136].

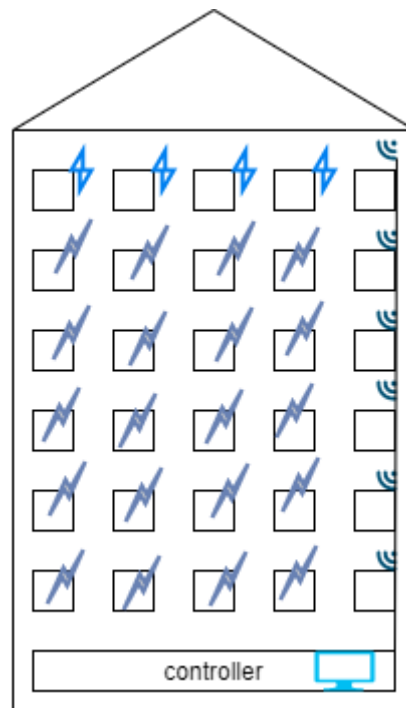


Fig 8.1. Smart Building

### 8.6. Related Work and Existing Technique

Several key schemes are proposed for IoT networks and sensor networks. Most of the schemes are lacking storage capacity computation power. In sensor network and Tot types, limited resources network key pre-distribution is mainly used due to traffic problems in the network. The first key redistribution scheme is proposed by the EG scheme given in [52]. The key is preloaded in nodes prior to deployment. A random key is selected from a big key pool. A network topology scheme is discussed in [96] offers a q-composite technique for nodes. In this scheme, nodes can send a secure message if they have q key common between them. But for large vulnerable networks, the technique becomes increasingly difficult to implement. Perrig and colleagues [137] created the SPINS, an old security protocol for the WSN. Key distribution center work performs by the base station, and two nodes may utilize the KDC to generate a pair of

keys for usage for neighbouring nodes. In [138], Wenliang DU et al. provide a matrix-based approach to creating secure keys. This technique depends on Bolm's private secret key established based on a matrix-based scheme, which was developed in 1984 [54]. Individual rows and columns act as keys for various nodes, resulting in a private and public matrix of equal size and form. This method is impossible to catch until a specific number of nodes has been reached, guaranteeing full resilience. This method is not scalable in the long term since the matrix is produced depending on the nodes within the network. Adding additional nodes to the network becomes challenging once a certain number of nodes has been deployed in the field. Zhu et al. used a hybrid method while developing their localized encryption and authentication protocol (LEAP+) to achieve this objective. The individual, pairwise, group, and network-wide keys are all available in this protocol. During the communication process, individual keys are exchanged between both the node and base station BS. In between the node and its neighbours, a pairwise key is exchanged. To minimize duplication of effort, a GroupWise key is utilized for clustered communication. A network-wide key shared by all nodes in the system protects the base station, responsible for broadcasting information throughout the network. Before deployment, each node is supplied with a unique key. A function is stored in every node that is used to generate individual key and other key. A location aware clustered based protocol is given in [56]. In this protocol location information is required to create the cluster and share the keys.

### **8.7. Proposed Scheme**

The suggested scheme addresses the important key setup for a heterogeneous network. We consider that all the nodes in the network have various capabilities, and clustered structure is already formed in the IoT network. A controller is there to control the activities. In this system, BS has a key pool that contains a large number of keys and their identity number. A function and seed value is distributed to every node before deployment.

#### **Network Model.**

We consider a multi-story building with many heterogeneous connected devices on different floors with different controllers at each level, as shown in figure 8.1. A centrally controlled server on the ground floor can send the data to every controller at

a different level. Keys are distributed to the heterogeneous device. Every device can be authenticated by a controller. The two-level controller can be used to reduce the traffic in the system.

### **Key Manager Hierarchy**

Main Controller works as the key distribution centre KDC and the cluster head works as KDC. Key revocation and setup are decentralized, and all the data authenticate by the central server. The proposed method handles key distribution by using a tree structure. This method reduces transmission costs by concurrently broadcasting multiple key-encryption keys. This scheme structure consists of devices located at the tree's leaf nodes and a central control centre is known as the KDC, which manages the virtual tree of keys. Each leaf node uses a secret key to interact with the KDC. The main controller is at the root of the tree, while the intermediate nodes are at CH. Each device in the leaf nodes within the same subtree is aware that each node belongs to an internal node rooted to a certain internal node.

---

#### **Algorithm 8.1: Algorithm Key Setup**

---

Step 1	:	A big pool of random numbers is generated by the main controller.
Step 2	:	Base controller BS randomly selects keys from the pool and stores them in the node
Step 3	:	Every node is gathered with a random number and formula
Step 4	:	After deployment, a node sends the message to the cluster head to join the network by sending the message $M = Id    seed    nonce$
Step 5	:	CH authenticates the node with the help of the main controller and allows the node to join the network
Step 6	:	A key is generated with the help of function and seed
Step 7	:	Exit

---

This technique, which is detailed below, is used to manage group communication across IoT device groups. Many users may subscribe to the same IoT device group, so it would be more efficient if the group key for encryption could be shared by all of these devices and their registered users. Keys are distributed to the heterogeneous device. Every device can be authenticated by a controller. The two-level controller can be used to

reduce the traffic in the system. Initially, we assume that based on controller BS and every CH node has a pair of public/private keys for authentication and secure communication. Every node knows the secret seed, and the function is known all over the network. The Scheme may be applied according to the key setup algorithm.

---

**Algorithm 8.2: Algorithm Key Revocation**

---

Step 1	:	An IDS system informs the main controller of the malicious activity of the node
Step 2	:	The Main controller informs the CH to refresh the key with some seed and the same formula.
Step 3	:	CH informs the nodes in the groups to generate a new key with a seed to an individual node.
Step 4	:	Nodes generate the key and inform the CH
Step 5	:	CH informs the main controller.
Step 6	:	Exit

---

**8.8 Security Framework for Heterogeneous Networks**

It is easy for the lifespan of a homogenous sensor network to be drastically decreased because a few or more nodes do much more activities than the rest of the network. Heterogeneous sensor nodes have been developed to address this problem. Node heterogeneity is classified into three types, connection heterogeneity, computation heterogeneity, and energy heterogeneity. Using heterogeneous nodes has several benefits, including improved lifespan and dependability and reduced data transfer latency. This research uses CH heterogeneous nodes with long-life batteries, which are used for aggregation, rapid data transfer, clustering, and a few security-related activities. Different than that, CHs are used for a variety of other applications. Furthermore, the CHs are equipped with GPS sensors, which allow them to track their exact location on the battlefield.

Modern networks like IoT and home controller networks are also prone to attack by different adversaries. Security of these networks is critical to function smoothly. These networks are heterogeneous networks with different types and capability devices. The Internet of Things (IoT) has the potential to connect billions of devices and services at

any time and from any location, with a wide range of applications. A heterogeneous network connects various technologies in an Always Best-Connected manner.

Whenever some security technique is implemented in these networks, we want two main requirements authentication and confidentiality. Data freshness and availability are also required as security requirements. For authentication, hash code and secret values are used, and for confidentiality, encryption and decryption are used. For any cryptography or hash method, secret keys are used. These keys are distributed to the node secretly and change or refresh over intervals. So, key management is an important task for large-scale networks. The scheme is given in the previous chapters. Scalable key management is given and implemented for homogenous networks. To implement this scheme in heterogeneous IoT networks, the scheme is modified. On the higher level, these networks differ in terms number of devices and types of devices. In modern IoT networks, the base station is replaced by the control centre. Sending secure data between devices is the main challenge in these types of networks.

User authentication and device authentication are two different scenarios, and both are involved in modern networks. Machine or device authentication requires a more precise algorithm and secure process. The above-presented scheme can work for hierarchal heterogeneous systems with some modifications. The hierarchical Internet of Things network is a subset of the general Internet of Things network comprised of several nodes organized in a hierarchy, such as the IoT gateway, cluster head nodes, and sensing nodes, among others. A user must be able to directly obtain actual data from sensor nodes for a particular application in a general Internet of the Things networking environment.

There are various IoT networks in the new research area, and trend in sensor technology that connects various sensors embedded in different delays uses things to make it easy to use and improve the performance of any system [139]. A body area network (BAN) is also the network of various wearable gadgets and medical equipment that keeps tracking the parameters of the human body like temperature, pressure, heartbeat, and haemoglobin, sugar laver. All the data is collected at some server and analysis automated by the machine or manually by doctors. According to data analysis, reports are generated, and treatment is given to the patient or advice is given to the client [140]. There are many challenges in IoT networks like security, QoS, routing of data, real-

time monitoring performance, and connection of heterogeneous devices. Limited battery power and limited range and bandwidth are of wearable gadgets is also a concern about these types of network. In this research, we also list applications and challenges related to these networks [141].

### **8.9. Security Analysis**

Unlike the vast majority of existing key systems, our method ensures that every connection is completely secured on a per-connection basis. While the likelihood of two adjacent nodes having the same key is less than .5, the possibility of two neighbouring nodes having the same key is much lower, with the probability being between 1 and 0. According to our methodology, the chances of this happening are always one in a hundred. Instead of a paired node or subgroup key produced by each communicator pair, a private key is used to establish a direct secure connection between two communicators. Following that, it provides an excellent connection. Network performance increases when the proposed method is used since packet loss, computational cost, energy usage, and time delay are all reduced, resulting in improved network efficiency. It was found that the usage of a single hop key management technique has decreased performance. The packet loss rate increased throughout the attack, resulting in many retransmissions of lost packets. The single-hop method was employed many times throughout the attack. Using a single-hop key for connecting with neighbour nodes and a multi-hop key for engaging with the multi-hop node at the terminal, the suggested combination method reduces traffic loss and energy usage. Compared to the other two techniques, the one-way function is responsible for the reduced computational overheads needed for a successful key generation operation.

### **8.10. Summary**

The number of wireless devices grows rapidly with each passing day, and the network's total size grows to an ever-increasing degree. Because smart gadgets are rapidly evolving, strict security requirements are needed to safeguard them from hackers. This article improves on earlier published work by presenting a hash-based key management technique for multi-hop networks. This paper presented a novel and adaptable Key Management system for the Internet of Things that secures both group and device-to-

device communication while being simple to deploy. To accomplish this objective, our approach is divided into three tiers. The top layer of the network is what divides the network into distinct groups. They are then assigned nodes depending on the services to which they have subscribed and the node's location. The nodes of each group are divided into logical subgroups by the middle layer, which is subsequently disseminated to the outer layers. Each of them must suffer overhead proportional to its members' skills. The aim is to distribute loads among heterogeneous devices in a way that is consistent with their capabilities. The hash function is essential because it aids in the security of stored passwords and data integrity. It is unnecessary to encrypt the whole data set with the secret key when using the recommended hashing technique; instead, encrypting just the hash value is sufficient for security reasons. Comprehensive security research was also conducted, which covered a broad range of desired security features. Furthermore, since it lowers the amount of storage, communication, and processing needed, our proposed approach beats the competition in terms of performance. Finally, a decentralized architecture increases scalability while reducing overhead for devices with low resources. We want to decentralize the protocol in the future via our efforts. Consequently, cryptographic material will be spread across numerous organizations to prevent a single point of failure and make accessing or altering this secret information more difficult. Further research for the complex mobile network may be continued, and a modified scheme may be used for IoT networks.

## **Chapter 9**

### **Conclusions and Future Research Scope**

#### **9.1 Overall Summary**

Key management is critical for network security in all situations. In addition to authentication and privacy, key management is essential for several other security aspects. Both computer networks and wireless networks, like our daily lives, are based on a basic concept that may be changed. A key is a secret which is not known to everyone such as only family members know their home's keys. Relevant parties must use encryption keys to keep private information concealed from the public eye while communicating. Keys are also used to identify people in a system who have been given access to certain information. Surveillance systems are used in various contexts, including environmental data gathering, industrial operations, and medical care administration. Furthermore, military operations in hazardous regions depend on these networks, and their safety is a key component of their operation. They need keys to guarantee data secrecy and to authenticate the cryptographic node methods necessary for system operation. The importance of credentials in hashing, decryption, and encryption processes cannot be overstated. Maintaining the keys' location over a large number of devices is indeed a time-consuming as well as difficult job. Key management is the process of generating and distributing keys for each node in a secure way. When you commit, you may also revoke keys from compromised nodes, and after a certain amount of time, you can give keys to a newly installed node that has been hacked. Key refreshments are essential for maintaining network security regularly, and they are also essential methods for key establishment. Key techniques, such as RSA and Diffie-Hellman, are unsuitable for sensor network settings due to their restricted processing and computing capacity. Because it is hard to meet all of an application's requirements, it is also difficult to decide which key management system is best suited to the application at hand. Several important control techniques have been suggested for dealing with large sensor networks. For key management, in WSN, many key schemes are proposed in the literature based on the matrix, polynomial, random, partial key, and tree-based. Some schemes are probabilistic and do not give a guarantee of connectivity of a complete network, and some are probabilistic schemes that give high connectivity, but more computation is required. Some schemes are not scalable and



cannot be used in large networks. This research proposed and analysed a new key management protocol for pairwise key establishment and key establishment for clusters networks. We also analysed how our key scheme may be used in modern networks like IoT. Our simulation results show that the scheme does better than existing techniques and supports key refresh and scalability with node addition support.

## **9.2 Conclusion**

Due to a shortage of resources, the security of the wireless network is a crucial and difficult task. Traditional key management in sensor networks is inefficient due to the huge demands on computing, storage, and energy resources. Due to the high volume of traffic in key distribution, key pre-distribution is utilized to disseminate keys. Before being distributed in the field, keys are stored in nodes as part of the pre-distribution process. The two most challenging problems to address are key renewal and scalability. In pre-distribution, connectivity and resilience are also important considerations.

According to the findings given in this research, selecting important parameters and values with care may assist in avoiding major attacks while simultaneously enhancing network connection quality. A technique for distributing keys in this system is shown that allows for more resilience to be obtained with a fewer number of key distributions. Small key pools are also utilized in certain cases. Using a key chain pool instead of a single large key pool increases node capture endurance by ensuring that the previous key cannot be calculated from the new key. Implementing a key management approach in a huge Wireless communication with grouped formations of sensor nodes is needed to deal with the high number of keys that exist in the system. Before this, asymmetric cryptography methods for wireless network security have been proposed, with the Diffie-Hellman key agreement and RSA cryptography being the most prominent examples. Owing to the limited computational and energy resources available to sensor nodes in non-wireless sensor networks, to address this problem, a key distribution technique based on the trusted server approach was created, which was then utilized to conduct asymmetric public key certification using public-key certification as the basis. Another alternative was the key pre-distribution method, which was extremely successful since it stores the information in nodes and allows for a more rapid deployment time. Following the key pre-distribution method proposed by Eschenauer and Gligor, the base station produces many random keys at random times. In other

words, if they are unable to generate a route key, they will be unable to interact with one another, which is analogous to not being able to locate the secret key.

Because in the proposed scheme, key chain length is maintained, employing a key chain pool rather than key pool give better results in terms of node capture resistance. Increased the length of the chain leads to improved results; nevertheless, as key chain length grows, more calculation is required, resulting in a reduction in the benefit gained from increasing. Key chain optimization is dependent on determining which hash chain produces the best results. This is a crucial component of key chain optimization. The value of pool size is also significant to consider since the likelihood of key sharing reduces as pool size decreases, and a large key pool is required for optimum key overlapping. Because sensor networks are very vulnerable, we have concentrated our efforts on increasing their resilience, which is one of the most significant factors affecting their security.

### **9.3 Future Research Scope**

Smart surveillance is made possible via the application of new sensors and IoT technology in the field of computer networks. Internet of Things networks, machine learning, and artificial intelligence are examples of such technologies. Future sensor node improvements will result in attractive and cost-effective devices for applications such as underwater acoustic sensor systems, sensing-based cyber-physical systems, time-critical applications, cognitive sensing and spectrum management, and security and privacy management. WSNs may be implemented in various locations, including on the road, underground, underwater, in woods, battlefields, disaster-prone regions, worksites, etc. Because of its pervasiveness, WSN is one of today's most pressing demands. WSNs may be used for smart cities, smart buildings, smart cars, and modern space technology. The security of any system is essential, and it should be done with care and precision. When it comes to trust in this quickly expanding world of interconnected gadgets, one system has faith in another device, as well as between a computer and a human. It is critical to understand the security difficulties presented by these systems and apply the security approach with care to ensure that the system is trustworthy. Efforts should be made to ensure that the two primary secrecy and message authentication issues are adequately addressed. Secret keys are used to maintain

authentication and confidentiality in these networks. In this research, we attempt to address all of the security problems that exist and offer the results of a survey on security challenges in the WSN and Internet of Things. The basic technique of random key pre-distribution is the most talked-about subject in the field of key management. The issue of safely updating the key at regular intervals is still under investigation. Most contemporary key management systems store cryptographic secrets in the device before they are installed in the field, allowing for faster deployment. We provide a technique for implementing security in static sensor network networks and improving connectivity and resiliency while maintaining network performance at the same level. This study is mainly concerned with static and large networks. This study may be expanded to include mobile sensor networks. We also developed a security framework for current IoT networks, which may be expanded to investigate these networks' security further. Some emerging IoT applications, such as smart homes, smart cities, and intelligent vehicles, may employ these schemes to increase network security, although additional study is necessary for mobile networks. Modern IoT networks feature many kinds and vendor's devices that were not designed with safety in mind when they were created. As a result, more research for heterogeneous devices may be done to implement and improve the security of these networks. There has been a steady shift from a general focus on the security of WSN, and they were either developed for grouped sensors or created for homogenous sensor networks. The requirement for key management systems that are unique to various application domains, such as healthcare applications and smart homes, is predicted to be a major focus of future research initiatives. An investigation of the use and performance of the distributed sensor network model under a variety of conditions, including the network's size, is required. It is possible to extend the transmission range to enhance the likelihood of secure communication between nodes. The energy cost, on the other hand, grows in direct proportion to the distance travelled. In the future, a new strategy that takes into account energy efficiency as well as secure communication will be studied. Symmetric key-based systems are frequently utilized because they have a low computational cost and are good for restricted resource sensors, which makes them particularly attractive. The shortages of symmetric key systems, on the other hand, are readily apparent. Distinct schemes may have different weaknesses in terms of security strength (resilience),

scalability, and the likelihood of a connection being made. On the other hand, public key systems provide several advantages, including lower communication overhead, more storage capacity, and greater scalability. It has the potential to deliver a simpler solution with significantly greater security strength. For wireless sensor networks, public key solutions were considered to be too costly in terms of processing. Furthermore, it is predicted that the processing cost would be far quicker than the cost of transmitting and receiving data. Apart from that, next-generation sensor nodes are planned to have ultra-low-power circuitry that will enable continuous energy delivery. As a result, with the rapid advancement of technology, public-key systems are inline and are expected to become extensively utilized soon.

## References:

- [1] X. Gong, H. Long, F. Dong, and Q. Yao, "Cooperative security communications design with imperfect channel state information in wireless sensor networks," *IET Wirel. Sens. Syst.*, vol. 6, no. 2, pp. 35–41, 2016, doi: 10.1049/iet-wss.2015.0003.
- [2] R. B. Agnihotri, A. V. Singh, and S. Verma, "Challenges in wireless sensor networks with different performance metrics in routing protocols," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*, 2015, pp. 1–5.
- [3] J. Wu, K. Ota, M. Dong, and C. Li, "A hierarchical security framework for defending against sophisticated attacks on wireless sensor networks in smart cities," *IEEE Access*, vol. 4, pp. 416–424, 2016.
- [4] J. A. Anastasov, A. M. Cvetković, D. M. Milović, D. N. Milić, and G. T. Djordjević, "On physical layer security in WSN over GK fading channels during intercept events," *Telecommun. Syst.*, vol. 74, no. 1, pp. 95–102, 2020.
- [5] R. Vijaya Saraswathi, L. Padma Sree, and K. Anuradha, "Dynamic and probabilistic key management for distributed wireless sensor networks," *2016 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2016*, 2017, doi: 10.1109/ICCIC.2016.7919666.
- [6] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *J. Netw. Comput. Appl.*, vol. 33, no. 2, pp. 63–75, 2010.
- [7] D. Liu, P. Ning, and W. Du, "Group-based key predistribution for wireless sensor networks," *ACM Trans. Sens. Networks*, vol. 4, no. 2, pp. 1–30, 2008.
- [8] A. Mohaisen, D.-H. Nyang, and T. AbuHmed, "Two-level key pool design-based random key pre-distribution in wireless sensor networks," *KSII Trans. Internet Inf. Syst.*, vol. 2, no. 5, pp. 222–238, 2008.
- [9] Y. Zhang, X. Li, J. Liu, J. Yang, and B. Cui, "A secure hierarchical key management scheme in wireless sensor network," *Int. J. Distrib. Sens. networks*, vol. 8, no. 9, p. 547471, 2012.
- [10] B. Elbhiri, R. Saadane, and D. Aboutajdine, "Stochastic and Equitable Distributed Energy-Efficient Clustering (SEDEEC) for heterogeneous wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 7, no. 1, pp. 4–11, 2011.
- [11] M. P. Đurišić, Z. Tafa, G. Dimić, and V. Milutinović, "A survey of military applications of wireless sensor networks," in *2012 Mediterranean conference on embedded computing (MECO)*, 2012, pp. 196–199.
- [12] Y. Zang, J. P. Thomas, G. Ghinea, M. Thomas, and S. Darwish, "Secure sector based bi-path clustering and routing protocol for WSNs," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 2012, pp. 777–784.
- [13] P. Goyal, M. Kumar, and R. Sharma, "A novel and efficient dynamic key management technique in wireless sensor network," *Int. J. Adv. Netw. Appl.*, vol. 4, no. 1, p. 1462, 2012.
- [14] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, 2009.

- [15] M. S. Hajar, M. O. Al-Kadri, and H. K. Kalutarage, "A survey on wireless body area networks: architecture, security challenges and research opportunities," *Comput. Secur.*, vol. 104, p. 102211, 2021.
- [16] M. Soni and D. K. Singh, "LAKA: lightweight authentication and key agreement protocol for internet of things based wireless body area network," *Wirel. Pers. Commun.*, pp. 1–18, 2021.
- [17] P. Majumdar, S. Mitra, and D. Bhattacharya, "IoT for Promoting Agriculture 4.0: a Review from the Perspective of Weather Monitoring, Yield Prediction, Security of WSN Protocols, and Hardware Cost Analysis," *J. Biosyst. Eng.*, pp. 1–22, 2021.
- [18] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557–3564, 2010.
- [19] A. Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla, "Using fire & ice for detecting and recovering compromised nodes in sensor networks," CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2004.
- [20] L. Haijun and W. Chao, "An energy efficient dynamic key management scheme based on polynomial and cluster in wireless sensor networks," *J. Conver. Inf. Technol.*, vol. 6, no. 5, pp. 321–328, 2011.
- [21] H. Choi, S. Zhu, and T. F. La Porta, "SET: Detecting node clones in sensor networks," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*, 2007, pp. 341–350.
- [22] A. C. Ferreira, M. A. Vilaça, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. Loureiro, "On the security of cluster-based communication protocols for wireless sensor networks," in *International Conference on Networking*, 2005, pp. 449–458.
- [23] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2–3, pp. 293–315, 2003.
- [24] H. Shafiei, A. Khonsari, H. Derakhshi, and P. Mousavi, "Detection and mitigation of sinkhole attacks in wireless sensor networks," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 644–653, 2014.
- [25] G. Jeong, Y.-H. Seo, and H. S. Yang, "Impersonating-resilient dynamic key management for large-scale wireless sensor networks," *Int. J. Distrib. Sens. Networks*, vol. 9, no. 6, p. 397926, 2013.
- [26] T. Giannetsos and T. Dimitriou, "LDAC: A localized and decentralized algorithm for efficiently countering wormholes in mobile wireless networks," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 618–643, 2014.
- [27] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," 2006.
- [28] A. S. Zahmati, B. Abolhassani, A. A. B. Shirazi, and A. S. Bakhtiari, "An energy-efficient protocol with static clustering for wireless sensor networks," *Int. J. Electron. Circuits Syst.*, vol. 1, no. 2, pp. 135–138, 2007.
- [29] N. Varela, O. B. Pineda Lezama, and H. Neira, "Information security in WSN applied to smart metering networks based on cryptographic techniques," *J. Intell. Fuzzy Syst.*, no. Preprint, pp. 1–8, 2020.

- [30] B. E. Manjunath and P. V Rao, "Balancing Trade off between Data Security and Energy Model for Wireless Sensor Network," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 2, p. 1048, 2018.
- [31] S. H. Erfani, H. H. S. Javadi, and A. M. Rahmani, "A dynamic key management scheme for dynamic wireless sensor networks," *Secur. Commun. Networks*, vol. 8, no. 6, pp. 1040–1049, 2015.
- [32] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning, "Defending against sybil attacks in sensor networks," in *25th ieee international conference on distributed computing systems workshops*, 2005, pp. 185–191.
- [33] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion detection systems in wireless sensor networks: a review," *Int. J. Distrib. Sens. Networks*, vol. 9, no. 5, p. 167575, 2013.
- [34] A. Martin and D. Khazanchi, "Information availability and security policy," 2006.
- [35] J.-Y. Huang, I.-E. Liao, and H.-W. Tang, "A forward authentication key management scheme for heterogeneous sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, pp. 1–10, 2011.
- [36] M. Boujelben, O. Cheikhrouhou, H. Youssef, and M. Abid, "A pairing identity based key management protocol for heterogeneous wireless sensor networks," in *2009 International Conference on Network and Service Security*, 2009, pp. 1–5.
- [37] B. Yahya and J. Ben-Othman, "REER: Robust and energy efficient multipath routing protocol for wireless sensor networks," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, 2009, pp. 1–7.
- [38] V. Goyal and G. Arora, "Review paper on security issues in mobile adhoc networks," *Int. Res. J. Adv. Eng. Sci.*, vol. 2, no. 1, pp. 203–207, 2017.
- [39] M. Oussalah, S. O. Amara, and R. Baghdad, "SECURING WIRELESS SENSOR NETWORKS: A SURVEY."
- [40] F. Bao, R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 9, no. 2, pp. 169–183, 2012.
- [41] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2003-Janua, pp. 197–213, 2003, doi: 10.1109/SECPRI.2003.1199337.
- [42] G. Han, X. Li, J. Jiang, L. Shu, and J. Lloret, "Intrusion detection algorithm based on neighbor information against sinkhole attack in wireless sensor networks," *Comput. J.*, vol. 58, no. 6, pp. 1280–1292, 2015.
- [43] F. Gandino, B. Montrucchio, and M. Rebaudengo, "Key management for static wireless sensor networks with node adding," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1133–1143, 2013.
- [44] M. E. Haque, N. Matsumoto, and N. Yoshida, "Context-aware cluster-based hierarchical protocol for wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 4, no. 6, pp. 379–386, 2009.
- [45] T.-P. Hong and C.-H. Wu, "An improved weighted clustering algorithm for determination of application nodes in heterogeneous sensor networks," 2011.
- [46] X. He, M. Niedermeier, and H. De Meer, "Dynamic key management in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 611–622,

- 2013.
- [47] U. M. Maurer and S. Wolf, “The diffie–hellman protocol,” *Des. Codes Cryptogr.*, vol. 19, no. 2, pp. 147–171, 2000.
  - [48] J. Kohl and C. Neuman, “RFC1510: The Kerberos network authentication service (V5).” RFC Editor, 1993.
  - [49] M. Ye, C. Li, G. Chen, and J. Wu, “EECS: an energy efficient clustering scheme in wireless sensor networks,” in *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, 2005, pp. 535–540.
  - [50] V. T. Kesavan and S. Radhakrishnan, “Cluster based secure dynamic keying technique for heterogeneous mobile wireless sensor networks,” *China Commun.*, vol. 13, no. 6, pp. 178–194, 2016.
  - [51] P. Hema, S. Sangeetha, R. K. Bora, and K. S. Rao, “PERFORMANCE ANALYSIS OF GAME THEORY FOR NETWORK SECURITY IN WSN.”
  - [52] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 41–47.
  - [53] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” in *IEEE INFOCOM 2004*, 2004, vol. 1.
  - [54] R. Blom, “An optimal class of symmetric key generation systems,” in *Workshop on the Theory and Application of Cryptographic Techniques*, 1984, pp. 335–338.
  - [55] C. W. Park, S. J. Choi, and H. Y. Youn, “A noble key pre-distribution scheme with LU matrix for secure wireless sensor networks,” in *International Conference on Computational and Information Science*, 2005, pp. 494–499.
  - [56] M. F. Younis, K. Ghumman, and M. Eltoweissy, “Location-aware combinatorial key management scheme for clustered sensor networks,” *IEEE Trans. parallel Distrib. Syst.*, vol. 17, no. 8, pp. 865–882, 2006.
  - [57] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, “An effective key management scheme for heterogeneous sensor networks,” *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.
  - [58] K. Lu, Y. Qian, M. Guizani, and H.-H. Chen, “A framework for a distributed key management scheme in heterogeneous wireless sensor networks,” *IEEE Trans. Wirel. Commun.*, vol. 7, no. 2, pp. 639–647, 2008.
  - [59] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-secure key distribution for dynamic conferences,” in *Annual international cryptology conference*, 1992, pp. 471–486.
  - [60] H. Dai and H. Xu, “Key predistribution approach in wireless sensor networks using LU matrix,” *IEEE Sens. J.*, vol. 10, no. 8, pp. 1399–1409, 2010.
  - [61] A. K. Das, “An efficient random key distribution scheme for large-scale distributed sensor networks,” *Secur. Commun. Networks*, vol. 4, no. 2, pp. 162–180, 2011.
  - [62] D. Du, H. Xiong, and H. Wang, “An efficient key management scheme for wireless sensor networks,” *Int. J. Distrib. Sens. Networks*, vol. 8, no. 1, p. 406254, 2012.
  - [63] W. Bechkit, Y. Challal, and A. Bouabdallah, “A new class of Hash-Chain based



- key pre-distribution schemes for WSN,” *Comput. Commun.*, vol. 36, no. 3, pp. 243–255, 2013, doi: 10.1016/j.comcom.2012.09.015.
- [64] E. A. M. Anita, R. Geetha, and E. Kannan, “A novel hybrid key management scheme for establishing secure communication in wireless sensor networks,” *Wirel. Pers. Commun.*, vol. 82, no. 3, pp. 1419–1433, 2015.
- [65] M.-L. Messai, H. Seba, and M. Aliouat, “A lightweight key management scheme for wireless sensor networks,” *J. Supercomput.*, vol. 71, no. 12, pp. 4400–4422, 2015.
- [66] Y. Zhang, J. Liang, B. Zheng, and W. Chen, “A hybrid key management scheme for WSNs based on PPBR and a tree-based path key establishment method,” *Sensors (Switzerland)*, vol. 16, no. 4, 2016, doi: 10.3390/s16040509.
- [67] F. Wu *et al.*, “An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment,” *J. Netw. Comput. Appl.*, vol. 89, pp. 72–85, 2017.
- [68] M. F. BAJESTANI and A. L. I. PAYANDEH, “A novel key distribution scheme against storage-bounded adversaries using attack probabilities,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1014–1021, 2016.
- [69] J. Choi, J. Bang, L. Kim, M. Ahn, and T. Kwon, “Location-based key management strong against insider threats in wireless sensor networks,” *IEEE Syst. J.*, vol. 11, no. 2, pp. 494–502, 2015.
- [70] F. Gandino, R. Ferrero, and M. Rebaudengo, “A key distribution scheme for mobile wireless sensor networks:  $\mathcal{S}_q$   $\mathcal{S}$ - $\mathcal{S}$  s  $\mathcal{S}$ -composite,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 1, pp. 34–47, 2016.
- [71] P. Ahlawat and M. Dave, “An attack resistant key predistribution scheme for wireless sensor networks,” *J. King Saud Univ. Inf. Sci.*, vol. 33, no. 3, pp. 268–280, 2021.
- [72] Y. Zhang and P. Li, “Key management scheme based on nodes capture probability for wireless sensor networks,” in *2018 Chinese Control and Decision Conference (CCDC)*, 2018, pp. 5470–5475.
- [73] M. L. Messai and H. Seba, “EAHKM+: Energy-aware secure clustering scheme in wireless sensor networks,” *Int. J. High Perform. Comput. Netw.*, vol. 11, no. 2, pp. 145–155, 2018, doi: 10.1504/IJHPCN.2018.10010950.
- [74] S. Athmani, A. Bilami, and D. E. Boubiche, “EDAK: An efficient dynamic authentication and key management mechanism for heterogeneous WSNs,” *Futur. Gener. Comput. Syst.*, vol. 92, pp. 789–799, 2019.
- [75] S. Rajasoundaran *et al.*, “Secure watchdog selection using intelligent key management in wireless sensor networks,” *Mater. Today Proc.*, 2021.
- [76] C. Iwendi, Z. Zhang, and X. Du, “ACO based key management routing mechanism for WSN security and data collection,” *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2018-Febru, pp. 1935–1939, 2018, doi: 10.1109/ICIT.2018.8352482.
- [77] M.-L. Messai and H. Seba, “EAHKM+: energy-aware secure clustering scheme in wireless sensor networks,” *Int. J. High Perform. Comput. Netw.*, vol. 11, no. 2, pp. 145–155, 2018.
- [78] K. Hamsha and G. S. Nagaraja, “Threshold cryptography based light weight key management technique for hierarchical WSNs,” in *international conference on ubiquitous communications and network computing*, 2019, pp. 188–197.

- [79] M. S. Yousefpoor and H. Barati, “DSKMS: a dynamic smart key management system based on fuzzy logic in wireless sensor networks,” *Wirel. Networks*, vol. 26, no. 4, pp. 2515–2535, 2020.
- [80] P. Alimoradi, A. Barati, and H. Barati, “A hierarchical key management and authentication method for wireless sensor networks,” *Int. J. Commun. Syst.*, p. e5076, 2021.
- [81] H. Delfs and H. Knebl, “Symmetric-key cryptography,” in *Introduction to Cryptography*, Springer, 2015, pp. 11–48.
- [82] A. Abduvaliev, S. Lee, and Y.-K. Lee, “Simple hash based message authentication scheme for wireless sensor networks,” in *2009 9th International Symposium on Communications and Information Technology*, 2009, pp. 982–986.
- [83] R. Bellazreg and N. Boudriga, “DynTunKey: a dynamic distributed group key tunneling management protocol for heterogeneous wireless sensor networks,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2014, no. 1, pp. 1–19, 2014.
- [84] H. Lee, Y. H. Kim, D. H. Lee, and J. Lim, “Classification of key management schemes for wireless sensor networks,” in *Advances in Web and Network Technologies, and Information Management*, Springer, 2007, pp. 664–673.
- [85] M. Abdelhakim, L. E. Lightfoot, J. Ren, and T. Li, “Distributed detection in mobile access wireless sensor networks under byzantine attacks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 950–959, 2013.
- [86] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.
- [87] K. Jiang, W. Wang, A. Wang, and H. Wu, “Network intrusion detection combined hybrid sampling with deep hierarchical network,” *IEEE Access*, vol. 8, pp. 32464–32476, 2020.
- [88] K. Shankar and M. Elhoseny, “Multiple Share Creation with Optimal Hash Function for Image Security in WSN Aid of OGWO,” in *Secure Image Transmission in Wireless Sensor Network (WSN) Applications*, Springer, 2019, pp. 131–146.
- [89] N. Tuah, M. Ismail, and K. Jumari, “Evaluation of optimal cluster size in heterogenous energy wireless sensor networks,” in *2012 International Symposium on Telecommunication Technologies*, 2012, pp. 124–130.
- [90] M. A. Simplicio Jr, 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,” *Comput. networks*, vol. 54, no. 15, pp. 2591–2612, 2010.
- [91] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastroianni, “Computer network simulation with ns-3: A systematic literature review,” *Electronics*, vol. 9, no. 2, p. 272, 2020.
- [92] D. Uma Vishweshwar, A. BalaRam, and T. Kishore Babu, “Secure Data Sharing Using Two Fold Cryptography Key Protection, Proxy Re-encryption and Key Separation Techniques,” in *ICCCE 2020*, Springer, 2021, pp. 299–305.
- [93] A. Aborujilah *et al.*, “Security Assessment Model to Analysis DOS Attacks in WSN,” in *International Conference of Reliable Information and Communication Technology*, 2019, pp. 789–800.
- [94] B. K. Mishra, M. C. Nikam, and P. Lakkadwala, “Security against black hole attack in wireless sensor network-a review,” in *2014 Fourth International*

- Conference on Communication Systems and Network Technologies*, 2014, pp. 615–620.
- [95] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, “A two-tier data dissemination model for large-scale wireless sensor networks,” in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 148–159.
- [96] Q. Mamun, R. Islam, and M. Kaosar, “Secured Communication Key Establishment for Cluster-Based Wireless Sensor Networks,” *Int. J. Wirel. Networks Broadband Technol.*, vol. 4, no. 1, pp. 29–44, 2015, doi: 10.4018/ijwnbt.2015010103.
- [97] Y. Zhang, B. Zheng, P. Ji, and J. Cao, “A Key Management Method Based on Dynamic Clustering for Sensor Networks,” *Int. J. Distrib. Sens. Networks*, vol. 2015, 2015, doi: 10.1155/2015/763675.
- [98] I. F. Kurniawan and R. Bisma, “Multiple sensing application on wireless sensor network simulation using NS3,” in *Journal of Physics: Conference Series*, 2018, vol. 947, no. 1, p. 12011.
- [99] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, “Detecting node replication attacks in wireless sensor networks: a survey,” *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 1022–1034, 2012.
- [100] A. Ghosal, S. Halder, S. Sur, A. Dan, and S. DasBit, “Ensuring basic security and preventing replay attack in a query processing application domain in WSN,” in *International Conference on Computational Science and Its Applications*, 2010, pp. 321–335.
- [101] Z. Cui *et al.*, “A hybrid blockchain-based identity authentication scheme for multi-WSN,” *IEEE Trans. Serv. Comput.*, vol. 13, no. 2, pp. 241–251, 2020.
- [102] I. T. Downard, “Simulating sensor networks in ns-2,” NAVAL RESEARCH LAB WASHINGTON DC, 2004.
- [103] J. Kůr, V. Matyáš, and P. Švenda, “Two improvements of random key predistribution for wireless sensor networks,” in *International Conference on Security and Privacy in Communication Systems*, 2012, pp. 61–75.
- [104] B. A. Bakr and L. Lilien, “A quantitative comparison of energy consumption and WSN lifetime for LEACH and LEACH-SM,” in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 182–191.
- [105] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.
- [106] L. Yu, N. Wang, W. Zhang, and C. Zheng, “Deploying a heterogeneous wireless sensor network,” in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 2007, pp. 2588–2591.
- [107] L. Ding and Z.-H. Guan, “Modeling wireless sensor networks using random graph theory,” *Phys. A Stat. Mech. its Appl.*, vol. 387, no. 12, pp. 3008–3016, 2008.
- [108] O. Yagan, *Random graph modeling of key distribution schemes in wireless sensor networks*. University of Maryland, College Park, 2011.
- [109] M. Eltoweissy, M. Younis, and K. Ghumman, “Lightweight key management for wireless sensor networks,” in *IEEE International Conference on Performance, Computing, and Communications, 2004*, 2004, pp. 813–818.
- [110] P. Balister, A. Sarkar, and B. Bollobás, “Percolation, connectivity, coverage and

- colouring of random geometric graphs,” *Handb. large-scale random networks*, pp. 117–142, 2008.
- [111] W. Z. Khan, M. Y. Aalsalem, M. N. B. M. Saad, and Y. Xiang, “Detection and mitigation of node replication attacks in wireless sensor networks: a survey,” *Int. J. Distrib. Sens. Networks*, vol. 9, no. 5, p. 149023, 2013.
- [112] S. Yurish, *Modern sensors, transducers and sensor networks*, vol. 1. Lulu. com, 2014.
- [113] M. Yu, K. K. Leung, and A. Malvankar, “A dynamic clustering and energy efficient routing technique for sensor networks,” *IEEE Trans. Wirel. Commun.*, vol. 6, no. 8, pp. 3069–3079, 2007.
- [114] O. Moh’d Alia, “Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm,” *Inf. Sci. (Ny)*, vol. 385, pp. 76–95, 2017.
- [115] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, “Comparing elliptic curve cryptography and RSA on 8-Bit CPUs,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3156, pp. 119–132, 2004, doi: 10.1007/978-3-540-28632-5\_9.
- [116] C. Wang, R. S. Batth, P. Zhang, G. S. Aujla, Y. Duan, and L. Ren, “VNE solution for network differentiated QoS and security requirements: from the perspective of deep reinforcement learning,” *Computing*, vol. 103, no. 6, pp. 1061–1083, 2021.
- [117] G. S. Shahi, R. S. Batth, and S. Egerton, “MRGM: an adaptive mechanism for congestion control in smart vehicular network,” *Int. J. Commun. Networks Inf. Secur.*, vol. 12, no. 2, pp. 273–280, 2020.
- [118] Y. Zhang, X. Li, J. Yang, Y. Liu, N. Xiong, and A. V Vasilakos, “A real-time dynamic key management for hierarchical wireless multimedia sensor network,” *Multimed. Tools Appl.*, vol. 67, no. 1, pp. 97–117, 2013.
- [119] F. Wu, T. Wu, and M. R. Yuce, “An internet-of-things (IoT) network system for connected safety and health monitoring applications,” *Sensors*, vol. 19, no. 1, p. 21, 2019.
- [120] A. Aziz, K. Singh, W. Osamy, and A. M. Khedr, “Effective algorithm for optimizing compressive sensing in IoT and periodic monitoring applications,” *J. Netw. Comput. Appl.*, vol. 126, pp. 12–28, 2019.
- [121] F. Olivier, G. Carlos, and N. Florent, “New security architecture for IoT network,” *Procedia Comput. Sci.*, vol. 52, pp. 1028–1033, 2015.
- [122] P. Schulz *et al.*, “Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture,” *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, 2017.
- [123] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future internet: the internet of things architecture, possible applications and key challenges,” in *2012 10th international conference on frontiers of information technology*, 2012, pp. 257–260.
- [124] R. Porkodi and V. Bhuvaneshwari, “The internet of things (IOT) applications and communication enabling technology standards: An overview,” in *2014 International conference on intelligent computing applications*, 2014, pp. 324–329.
- [125] S. H. Shah and I. Yaqoob, “A survey: Internet of Things (IOT) technologies,

- applications and challenges,” in *2016 IEEE Smart Energy Grid Engineering (SEGE)*, 2016, pp. 381–385.
- [126] A. M. Vilamovska, E. Hattziandreu, R. Schindler, C. Van Oranje, H. De Vries, and J. Krapelse, “Rfid application in healthcare—scoping and identifying areas for rfid deployment in healthcare delivery,” *RAND Eur. Febr.*, p. 26, 2009.
- [127] H. El-Sayed, A. Mellouk, L. George, and S. Zeadally, “Quality of service models for heterogeneous networks: overview and challenges,” *Ann. Telecommun. des télécommunications*, vol. 63, no. 11, pp. 639–668, 2008.
- [128] M. Frustaci, P. Pace, G. Aloï, and G. Fortino, “Evaluating critical security issues of the IoT world: Present and future challenges,” *IEEE Internet things J.*, vol. 5, no. 4, pp. 2483–2495, 2017.
- [129] S. Yoon, H. Park, and H. S. Yoo, “Security issues on smarthome in IoT environment,” in *Computer science and its applications*, Springer, 2015, pp. 691–696.
- [130] H. Cai, B. Xu, L. Jiang, and A. V Vasilakos, “IoT-based big data storage systems in cloud computing: perspectives and challenges,” *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, 2016.
- [131] A. H. Sodhro *et al.*, “Quality of service optimization in an IoT-driven intelligent transportation system,” *IEEE Wirel. Commun.*, vol. 26, no. 6, pp. 10–17, 2019.
- [132] R. Sutaria and R. Govindachari, “Making sense of interoperability: Protocols and Standardization initiatives in IOT,” in *2nd International Workshop on Computing and Networking for Internet of Things*, 2013, p. 7.
- [133] L. Daniele, M. Solanki, F. den Hartog, and J. Roes, “Interoperability for smart appliances in the IoT world,” in *International Semantic Web Conference*, 2016, pp. 21–29.
- [134] C. Schmitt, M. Noack, and B. Stiller, “TinyTO: Two-way authentication for constrained devices in the Internet of Things,” in *Internet of Things*, Elsevier, 2016, pp. 239–258.
- [135] H. V. Nguyen and L. Lo Iacono, “REST-ful CoAP message authentication,” in *2015 international workshop on secure Internet of Things (SIoT)*, 2015, pp. 35–43.
- [136] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov, “Preparation, detection, and analysis: the diagnostic work of IT security incident response,” *Inf. Manag. Comput. Secur.*, 2010.
- [137] O. Delgado-Mohatar, A. Fúster-Sabater, and J. M. Sierra, “A light-weight authentication scheme for wireless sensor networks,” *Ad Hoc Networks*, vol. 9, no. 5, pp. 727–735, 2011.
- [138] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, 2005.
- [139] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, “Body area network security: a fuzzy attribute-based signcryption scheme,” *IEEE J. Sel. areas Commun.*, vol. 31, no. 9, pp. 37–46, 2013.
- [140] V. Mainanwal, M. Gupta, and S. K. Upadhayay, “A survey on wireless body area network: Security technology and its design methodology issue,” in *2015 international conference on innovations in information, embedded and communication systems (ICIIECS)*, 2015, pp. 1–5.

- [141] S.-Y. Chang, Y.-C. Hu, H. Anderson, T. Fu, and E. Y. L. Huang, “Body Area Network Security: Robust Key Establishment Using Human Body Channel.,” in *HealthSec*, 2012, p. 5.

---

## LIST OF PUBLICATIONS

### Published Papers

1. Vipin Kumar, Navneet Malik, Gaurav Dhiman, Tarun Kumar Lohani, "Scalable and Storage Efficient Dynamic Key Management Scheme for Wireless Sensor Network", *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5512879, 11 pages, 2021. <https://doi.org/10.1155/2021/5512879>
2. V. Kumar and N. Malik, "A Survey of Key Management Schemes for Large Scale Wireless Sensor Networks," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 628-631, doi: 10.1109/ICACCS51430.2021.9441849.
3. V. Kumar and N. Malik, "Dynamic Key Management Scheme for Clustered Sensor Networks with Node Addition Support," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021, pp. 102-107, doi: 10.1109/ICIEM51511.2021.9445393.
4. Kumar, V., Malik, N. Enhancing the connectivity and resiliency of random key pre-distribution schemes for wireless sensor network. *Int J Syst Assur Eng Manag* (2021). <https://doi.org/10.1007/s13198-021-01265-x>
5. Vipin Kumar, "Mobile Cloud Computing Security Challenges and Method: A Review" *IOSR Journal of Computer Engineering (IOSR-JCE)* 2019, 21(2) (pp. 12-16)

### Accepted Paper.

1. Vipin Kumar, Navneet Malik "Dynamic Group Key Management Technique in Context of Modern IoT Applications" 5TH INTERNATIONAL CONFERENCE ON COMPUTING SCIENCES (ICCS-2021), Lovely Professional University. 3-5TH DECEMBER 2021