Jimma University

Jimma Institute of Technology

School Of Computing

Torching Detection for Node Replica Attack In Cluster-Based Wireless Sensor Networks

By
Baessa Kajela

Co-Advisor:
Mr. Zelalem Hailu
(MSc.)

Advisor:
Dr. Tofik Jemal(PhD)

April 19, 2018

Jimma University

Jimma Institute of Technology

School Of Computing

Torching Detection for Node Replica Attack In Cluster-Based Wireless Sensor Networks

A Thesis Submitted To The School Of Computing Of Jimma University In Partial fulfillment for the degree of masters of science in computer Networks

By
Baessa Kajela

Co-Advisor:
Mr. Zelalem Hailu
(MSc.)

Advisor:
Dr. Tofik Jemal(PhD)

April 19, 2018

Jimma University

Jimma Institute of Technology

School Of Computing

Torching Detection for Node Replica Attack In Cluster-Based Wireless Sensor Networks

Name and Signature of the Examining Board Members:

Name	Signature	Date
1) Dr· Towfik Jemal,Advisor	.....	.....
2)	.....	.....
3)	.....	.....

April 19, 2018

Declaration

I hereby declare that this Master thesis entitled **"Torching Detection for Node Replica Attack In Cluster-Based Wireless Sensor Networks"** was carried out by me the degree of Master of Science in Computer Network under the guidance and supervision of Dr.Towfik Jemal , Jimma Institute of Technology, Jimma University, Ethiopia.

Except where specific reference is made to the work of others, the work presented in this thesis is based on my reading and understanding of the original texts and they are not published in whole or in part for consideration anywhere in the form of books, articles or monographs. The other books, articles ,and websites, which I have made use of are acknowledged at the respective place in the text.

For the present thesis, which I am submitting to the University, no degree or diploma or distinction has been conferred on me before, either in this or in other University. Thus, thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgement.

Baessa Kajela	Date

This thesis has been submitted for examination with my approval as university advisor

Towfik Jemal, PhD	Date

Acknowledgement

The one above all, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this thesis study and to preserve and complete it satisfactorily.

I wish to express my sincere thank to my advisor, Dr.Towfik Jemal for his real supporting and guidance from the beginning to the end of the study. He has been there providing his heartfelt support and guidance at all times and has given me invaluable guidance, inspiration ,and suggestions in my quest for knowledge. Without his able guidance, this thesis would not have been possible and I shall eternally be grateful to him for his assistance.

It is an honor for me to express my sincere thank all those who have contributed in one way or another to this study. Besides, I wish to express my sincere gratitude to all my classmates for their collaboration with giving me supportive ideas, comments and also for their encouragements.

I have great pleasure in acknowledging my gratitude to Mr.Amanuel Chali, Ph.D. student at Belgium University, for his incessant inspiration, directions, guidance, suggestions and, above all, his moral support.

Table of Contents

List of Acronyms

6LoWPAN	IPv6 over Low-power Wireless Personal Area Network
ADCs	Analog to Digital Converters
BS	Base Station
CBWSN	Cluster-Based Wireless Sensor Networks
CH	Cluster Head
CM	Cluster Member
CM	Cluster Member
CPU	Central Processing Unit
FFD	Full-Function Device
GPS	Global Positioning Systems
HART	Highway Addressable Remote Transducer
LEACH	Low Energy Adaptive Clustering Hierarchy
LRWPAN's	Low rate wireless personal area networks
MAC	Medium Access Control
PAN	Personal Area Networks
PDA	Personal Digital Assistant
PHY	Physical Layer
RFD	Reduced-Function Devices
SMP	Sensor Management Protocol
SQDDP	sensor query and data dissemination protocol
SQTL	sensor query and tasking language
TCP	Transport Control Protocols
TRA	Timed Release Algorithm
U-LEACH	Universal Low Energy Adaptive Clustering Hierarchy
UDP	User Datagram Protocol
WPANs	Wireless Personal Area Networks
WSNs	Wireless Sensor Networks

List of Algorithms

List of Figures

List of Tables

Abstract

Clustered Wireless Sensor Networks consists various sensor nodes with different roles. In this type network, CH is a vital sensor node which plays a local data aggregator role and delivery of the aggregated data to the BS. For that reason, the CH election should be protected from the adversary. Because the adversary can compromise the sensornode and creates the clones of the compromised sensor nodes. And finally, involve them in the CH election process to assign the CH role to the clone nodes. Many schemes have been proposed to protect the clone attack from being elected as CH during cluster formation operation, however, they suffer in case of communication cost and energy consumption. In fact, if process the CH election and rotates among trustworthy nodes, we can secure the cluster formation process from the clone attack. That is the reason we proposed a clone attack detection scheme which detects a clone attacks during CH election operation. The proposed scheme computes TRAPDOOR for each sensor node to control the release time of CH advertisement messages and identifies the exact cluster community that one sensor node should belong to it. The clone attack detection is conducted at the Basestation and the cluster community. Likewise, the performance and security analysis indicated that our scheme outperforms a clone attack detection during CH election process with low communication cost and energy consumption.

Index Terms

WSN, Cluster formation, Secure CH election against Clone attack, Clone attack

1. Introduction

1.1. Background

Nowadays the advances in wireless communication have increased the demands of low cost, low power, multi-functional sensor nodes in ubiquitous and pervasive applications [3]. Wireless Sensor Networks (WSNs) consist of large amounts of constrained sensor nodes with sensor unit, information processing unit, transceiver,storage resource and power supply components which convenient to Wireless Sensor Network's enabling technologies. Thus, WSNs are the communication paradigms with the collaboration of tiny sensor nodes. The tiny sensor nodes collaborate among themselves to set up a network and collecting, processing, analyzing, and disseminating data in multi-hop away. Then provide and routing accessed data to the BS over wireless communication. In WSNs, all information coming from sensor nodes is downstream to the BS.

Wireless Sensor Networks (WSN) are foreseeable to be the solution to many spectrums of applications such as military monitoring application, earthquake monitoring, tornado monitoring, health care, academic purpose, controlling traffic flows on the road [4]. WSN applications are engaging for information collecting to enhance help control and monitor of such applications. Thus, the possibilities of WSN applications are ranging from traffic monitoring to high-security applications with several phenomena applications. Due to limited resources of sensor nodes, WSNs might barrier to incorporates and achieves its potential function in some spectrum of applications. Particularly, energy is considered as a crucial resource in WSNs since most sensor nodes are battery operated i.e. their rechargeable devices. Once they are deployed, it is very difficult and challenge to replace or recharges their battery. Moreover, most of the time the operation demands of WSNs are based on the energy level of energy-constrained sensor nodes. Thus, these constraints might present the design challenges of WSNs in different computing applications.

To reduce these constraints and to achieve their potential functions, a novel mechanism and techniques are required for WSNs that take into consideration the constraints of sensor nodes such as energy consumption, open communication architecture, self-organizing and etc. Thus, node clustering is a novel technique, which partitions the WSNs into clusters, is critical saving energy consumption of sensor nodes and extending the lifetime of networks consequently [5]. Cluster formation involves three parties: Base Station (BS), CH and constrained sensor nodes. In many cases, CH is elected among the sensor nodes deployed in the application area. Once elected, CH act as a local aggregator to cluster being elected for. It gathers data from its Cluster members and forwards the aggregated data to the BS. By taking these into consideration, an adversary will distort the cluster formation to being elected as a CH.

Furthermore, WSNs constrained with self-organizing, unattended in nature, deployed often in hostile and physically insecure environments will open the way as an adversary has disrupted the successful completion of CH election. Beside to network nature constraints,

limited resources such as small memory or storage size, computational processing, battery power based operating and multi-hop away transmissions of sensor nodes will a security issue of this process. The critical network security of WSNs is that sensor nodes can easily vulnerable to physical capture attacks. The adversary can compromise legitimate nodes, clones and deploy in several positions of networks. Thus, the clone nodes will destroy and launch numerous types of internal attacks. Then, the networks will susceptible to many types of attacks such as reformation of the cluster,the malicious nodes becomes Cluster Head (CH) and consequently jamming signal, inject false information, including the intention to acquire a critical information, overrunning networks that result in the quickly depleting of sensor node's battery[6].

Node replica attacks are a serious security problem and destructive attacks in WSNs. In these types of attacks, adversary captures one or more legitimate node physically and compromises all its secret credential information and creates the clones or a replica of the compromised nodes and then deploying them within the network as shown in Figure 1.

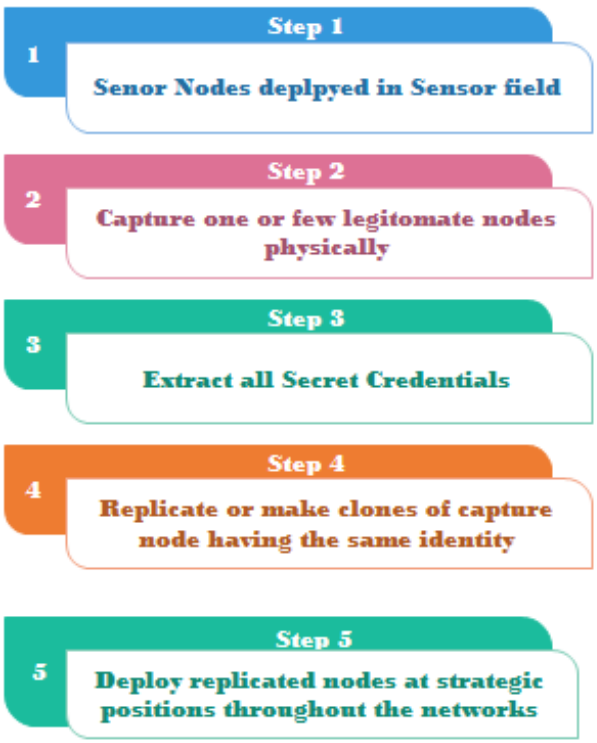

Figure 1. Node Replication Steps

These replicated nodes can participate in the cluster formation operation of networks in the same way as a legitimate node since they have legitimate access to the networks (keys, legitimate IDs, security credentials, others) nodes and characterized by :

- Node replica attacks act as the trustworthy node with its neighbourhood

without global knowledge.

- The legitimate node could not aware that there is a node replica attack among its neighbors.

Therefore, if these node replicas are not detected efficiently, the adversary will protect the proper election of CH and unsuccessful completion of the cluster formation process among legitimate node. Thus, attackers can take control over the network by exploiting these replicated nodes. Thus, node replica attacks are the most destructive attack in WSNs, the efficient security solution needed for node replicas detection to limit and mitigate their damage.

This observation motivates our research work on detecting and mitigate against node replica attacks during cluster formation of Wireless Sensor Networks (WSNs).

1.2. Statement of Problems

Energy is expensive resource signifies the life presence of Wireless Sensor Networks (WSNs). WSNs consists of a numerous number of sensor nodes deployed over a hostile area of targets to be observed. These sensors are battery operated, requires a power to carry out an assigned task demands and perform the network operations. After sensing information from the target area, sensor nodes transmit sensed data back to the BS using other nodes as a relay. Note that data obtained from one node are more associated with its neighbor's data, duplicated data transmitted over the network can be reduced through data aggregation of collaborated sensor nodes. Thus, one important aspect of sensor nodes collaboration in WSNs is that providing the trade-off between computational energy and communication cost.

Thus, the operation lifetimes of WSNs and functional life of sensor nodes are based on the demands request placed on its battery. All sensor nodes require energy power for transmitting the data over networks. In many applications, sensor nodes establish direct communication with the BS. Particularly when the BS far away from the deployment area, sensor node consumes large energy resource to deliver the sensed data back to the BS. Due to these challenges, the energy power of sensor nodes can drain out quickly and consequently the shortage of the WSNs lifespans. Instead of direct communication, using local data aggregation can be a significant solution for these problems.

Node Clustering is a novel technique that aggregates WSNs into small groups (clusters) based on their communication ranges. In Wireless Sensor Networks (WSNs), maintaining node clustering is very important for two reasons:

❶ Preserving and reducing the necessary energy consumption of Sensor nodes

❷ Consequently, enhancing network lifetime and performance.

In Clustered WSNs, in each cluster one sensor node is elected as CH once the sensor nodes are organized into clusters. The main idea is to organize sensor nodes around Cluster Head (CH) that is responsible for gathering data from its cluster members and forwards the gathered data back to the BS. In the other words, CH is a local aggregator for its cluster.

For that reason, one of the most important aspects should be considered in clustered WSNs is its security. An adversary can compromise the legitimate nodes, clones and deploy in the sensing area. The clone nodes can participate in the course of cluster formation and prepare themselves as CH candidates for being elected as CH. Then it might trick the number sensor nodes to joining a malicious Cluster Head to achieve the adversary roles. Beside of this, using clone nodes, an adversary can rotate the role of legitimate CH before role round terminated to divert the CH roles to the clone nodes. Using a very strong radio signal, a clone attack can transmit a fake Cluster-Head(CH) volunteer messages to become a Cluster Head and then adversary might deceive a number of sensor nodes into joining a not existed cluster.

Therefore, elected CH among trustworthy nodes and assigning a CH role to the eligible node has an observation motivates our work on designing resilient protocol on node replica attacks detection and mitigates at cluster formation.

1.3. Objectives

The overall objective of this study designing and developing a scheme for detecting and mitigating node replica attacks during the cluster formation operation of Wireless Sensor Networks (WSNs) .

1.4. Specific Objectives

In order to achieve the general objective stated above, the following specific objectives will be taken into account during this study:

- To review related research works in the area of node replica attack in Wireless Sensor Networks.
- To review techniques and protocols to detect node replica attacks in real times.
- To review the characteristics of node replica attacks on selected area.
- Electing and assigning Cluster Head (CH) to an only legitimate node.
- Rotates a CH role among legitimate node members.
- To expel node replica attacks from CH candidates.

- To design a detection protocol that detects node replica attacks throughout cluster formation and revoke the clone nodes among neighbor legitimate nodes afterward.
- To develop a detection protocol's prototype for the purpose of examining the effectiveness of designed scheme on node replica attacks detection.
- Evaluate and report experimental results founds
- And elicit the conclusion from acquired experiment results and recommend further research direction works in the area.

1.5. Methodology

The emphasis of this proposed approach is designing and developing the security solution to detect and mitigate the node replica attack in wireless sensor networks. To accomplish the aforementioned objectives of this study, several approaches take into account as methods:

1.5.1. Literature Review

To accomplish the objectives of this research work mentioned above, several articles, journal papers, magazine papers ,and literature were reviewed. Resources specifically related to the area of research effort on several types of attacks on Wireless Sensor Network were also reviewed to explore the best security solution to detect node replica attack in WSNs.

1.5.2. WSN topology

In WSNs, topology is limited by centralized (Cluster-Based) and distributed architectures. In a centralized architecture, WSNs hierarchical to several cluster structures. This approach has been applied to WSNs to network performance enhancements while reducing extra energy consumption. Beside to energy efficiency, Cluster-based architecture(Centralized architecture) will satisfy the following features in WSNs [7]:

- Achieving Network scalability.
- Reducing routing table stored as individual by localizing the route setup within the cluster.
- It can limit the scope of inter-cluster interactions to CHs and avoid redundant message exchanges between sensor nodes, which can also result in the conservation of bandwidth.
- In Cluster-Based architecture, the CH can create a transmission schedule for the sensor nodes within the clusters so that individual nodes switch to low-power sleep mode during transmission idle.

- A CH can reduce the number of relays packets by aggregating data collecting by sensor nodes in its cluster.
- Finally, CH can prevent medium access collision by engaging sensor nodes in a round-robin order and then determine the time for their transmission and reception of packets.

Clustered WSNs involve three components: Base Station, Cluster Head ,and sensor nodes as depicted in Figure 2.

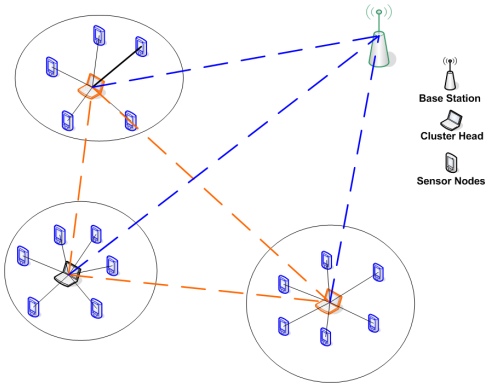

Figure 2. Architecture of Cluster-Based WSNs

❶ **Base Station**

The Base station is the most powerful and trusted agent act as the brain of WSNs. It performs monitoring activities such as communication between sensor nodes, adding or removing sensor nodes, detects malicious nodes, monitoring poll the status of each sensor node and etc..

❷ **Cluster Head**

It is a more powerful node selected from the collection of nodes. It provides better help monitor and controls the activities of sensor nodes with Base Station. And also, Cluster head keeps track of each sensor nodes and sends the status information of sensor nodes to the Base Station periodically.

❸ **Sensor Nodes**

These include normal sensor nodes deployed for sensing data in WSNs. They have been monitored by Cluster head and the Base Station in order to secure the operation carried out between them and check their current status.

By considering the importance of this architecture, this study was designed the resilient detection protocol in Cluster-Based WSNs for the safety cluster formation. Thus, our detection protocol detects node replica attacks in central WSN architecture by investigating its security issues.

1.5.3. Evaluation of Proposed scheme

The main contribution of this research works is designing and developing security mechanism for node replica attack and evaluate the effectiveness of the scheme in terms of communication overhead, Energy consumption, detection probability and Network lifetime.

1.5.4. Designing and Implementation

The models and algorithms of the proposed scheme will be designed in the design phase. Due to the convenient issue, the simulation of the proposed scheme will be conducted by OMNET++ in order to make sure that the effectiveness of scheme for detection of node replica attacks and evaluate its performance based on the evaluation metrics mentioned above. And also we use LaTeX for document scripting. Furthermore, the methodology of the proposed scheme is depicted in figure 3.

1.6. Scope and Limitation of the Study

The scope of this study is limited to designing and simulating the security mechanism to detect node replica attacks in Cluster-Based Wireless Sensor Networks. The proposed scheme consists of two phases: Cluster formation and node replica attack detection. During cluster formation, the scheme partitions the network into clusters. In the second phase, the scheme detects and revokes the compromised nodes try to become a Cluster Head. The scheme detects also a cloned node in the course of CH role rotation. The proposed scheme also designed cluster protocol for optimal node clustering and, significantly, CH election and rotation, which satisfies better node replica attacks detection. Furthermore, this study will aim to protect clone nodes to participate in the cluster formation process and assure CH election and rotation among legitimate nodes. **Detection, Revocation ,and Cluster_Formation** are the main course of actions in this scheme. In general, the proposed scheme limited to the following tasks:

❶ Clustering nodes based on their communication range, connectivity ,and energy level.

❷ Detect and revoke node replica attacks from CH candidates.

❸ Elect and Assign optimal CH in each cluster.

❹ Rotate CH role by the legitimate node.

❺ Finally formulating Cluster formation among trustworthy sensor nodes in Cluster-Based WSNs.

Furthermore, the proposed scheme will exclude the following tasks:

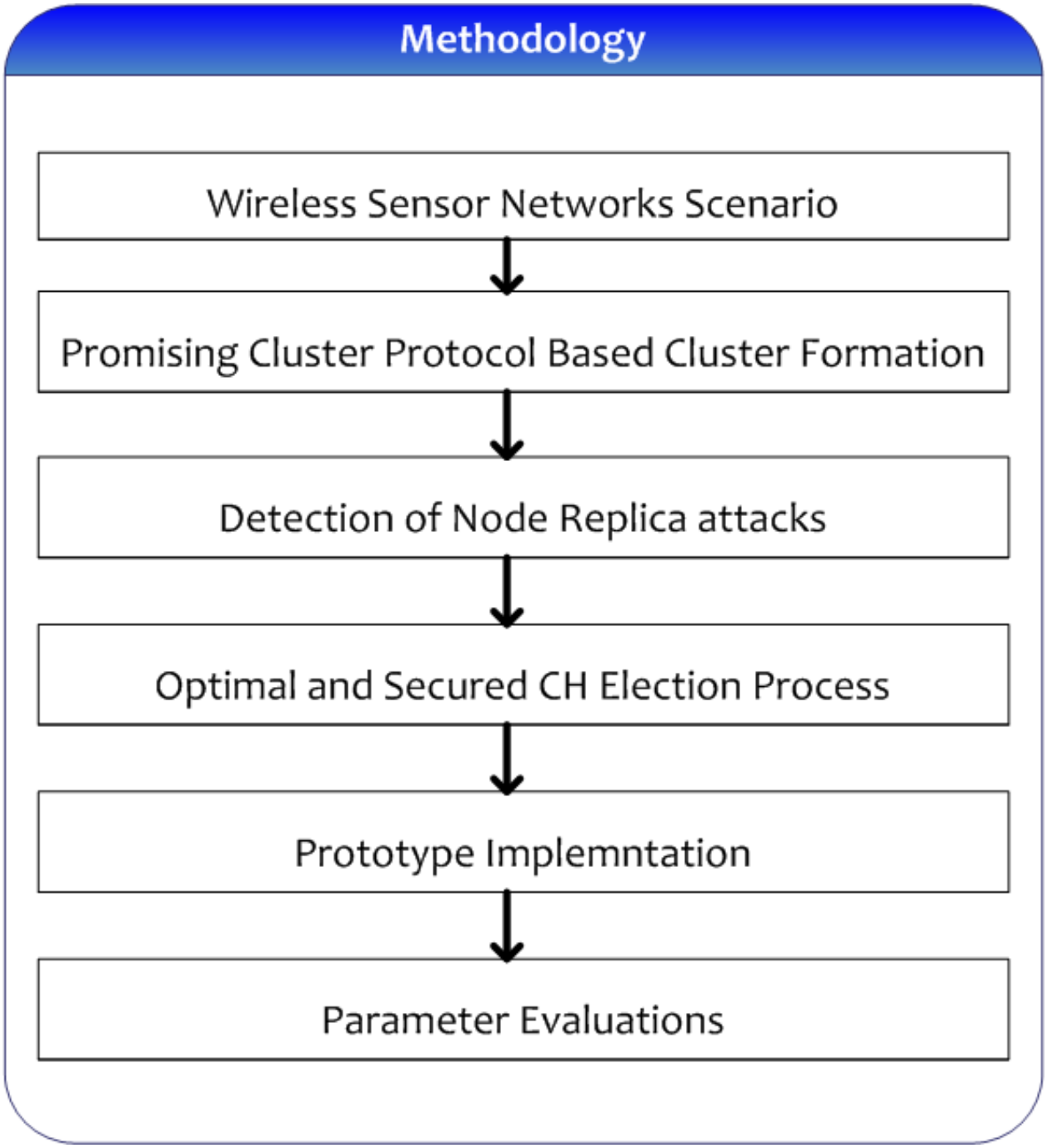

Figure 3. Methodology

❶ Other types of attacks such as sinkhole and related attacks are beyond the scope of this study.

❷ Detecting node replica attacks in distributed WSNs.

1.7. Contributions

The main contributions of this thesis are:

1) A novel energy balance node replica attack detection scheme is proposed and exten-

sively evaluated theoretically and in a simulation. Node replica detection scheme was designed based on the combination of cluster formation and node replica detection. It shows a promising performance compared to other competing detection scheme in-terms of communication cost, detection probability, prolonging network lifetime and energy consumption per packet.

2) The results were promising and the scheme exhibited the desired properties: gelocalizing sensornodes, cluster formation, optimal routing solution for minimizing energy consumption, optimal node replica detection during cluster formation, and significantly prolongs the network lifetime. In summary, the node replica attack detection scheme designed in thesis not only support node replica detection requirements, but also able to minimize energy consumption in sensornodes.

1.8. Organization of the Study

This study is organized into six chapters. Chapter One presents out the background of the study, statement of problem, objectives of the study, methodology as well as the limitation and delimitation of the study. Various states of arts of WSNs: applications, standards, technology node architecture, network architecture, security issues and sensor node constraints are discussed in Chapter Two.

Chapter Three deals with several works done related to node replica attacks and cluster formation in WSNs.

Chapter Four concerns with the description of models and algorithms designed based on the proposed scheme objectives.

In Chapter Five, the simulation and evaluation of the proposed scheme are discussed. The chapter also discusses the analysis of the results found from simulation, using evaluation metrics discussed in Section 1.5.3.

In Chapter 6, the conclusions of the proposed scheme in thesis are summarized and some suggestions for future work are presented.

2. Literature Review

The computing devices are involved the different evolution of paths to becoming the inseparable part our everyday life. We envision that this can happen that by reducing the physical appearance of computing devices and provide them to access computing capability over a broadband network using lightweight devices. This evolution introduces that the movement of the computer from insulated and sealed rooms to body computing devices. With the introduction of computer networks and advances in micro-electro-mechanical systems, a new technology paradigm has emerged since the last decades so-called ubiquitous computing. Particularly the aim of this paradigm is to make many computers to be available throughout the physical environment but making them effectively hidden from the user. Recent advances in wireless communication and digital electronics have enabled the production of low cost, low power and smart devices such as smartphones,PDA, Radio Frequency Identification Systems, Wireless Sensor Networks and many other computing technologies [8, 9].

The remaining sections of this chapter are intended to present: the introduction, transmission median, security issues, applications, enabling technologies and other related issues of Wireless Sensor Networks.

2.1. Overview of Wireless Sensor Networks

Wireless Sensor Networks consist a collection of sensor nodes, each equipped with storage, one or more sensors, power, transceiver or communicating components, processing resources and sometimes positioning systems like Global Positioning Systems (GPS). This sensor device has the capability to sense the environmental targets (rescue, tornado, humidity,etc.) and also they able carry out simple computation like sensed/received data and communicate with other nodes within communication range. Since they have no predetermined deployment architecture, they deployed in the specified area constitutes a network with no pre-established architecture and achieve the objective sensor networks with the help of the collaborative effort of a large number of nodes to compute and collect data in a very efficient way.

The emergence of a WSNs paradigm triggers many research areas and deliver many applications of WSNs in different fields that can realize the using of WSNs [1]. Some of the application areas are physiological data of patients, military monitoring application, building surveillance, fire protection and so on. To enable Wireless Sensor Networks (WSNs) applications using sensor technologies, the range of tasks can be classified in three as shown in Figure 4 [2]. The first class is a system which represents each individual sensor node. These devices require the development of new platforms, operating system ,and storage. The second class is communication protocols, which enable the communication

application and sensors and between sensor nodes. And the last class is a service which is designed to enhance the application and improve system performance and network efficiency [2]. The possible applications of WSNs are further discussed in Sect. 2.1.1.

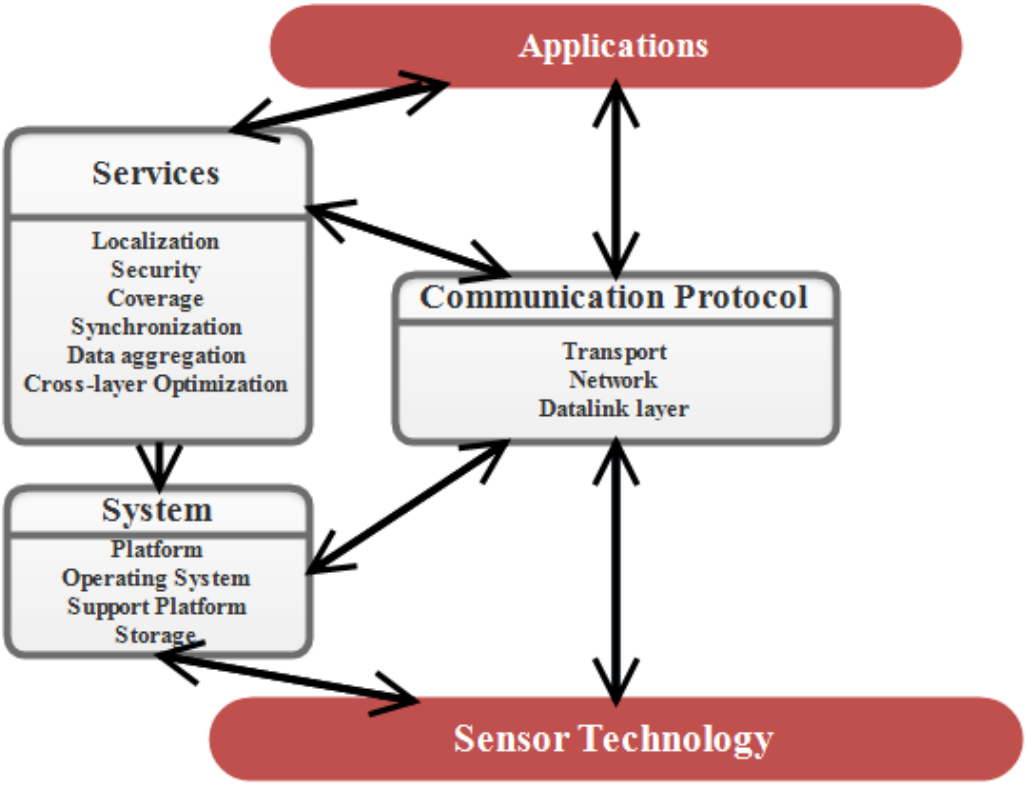

Figure 4. Classification of various issues in a WSNs

2.1.1. WSN Applications

Wireless Sensor Networks (WSNs) consist of several types of sensors, including thermal,visual, radar and infrared, which are enabled WSNs to monitor varies ranges of applications like pressure, rescue operation supporting, sniper detection and so on [1]. But recently, the interest of Wireless Sensor Networks (WSNs) Applications gains more focusing on networked biological and chemical sensors for national security applications. Potential possible applications of Wireless Sensor Networks (WSNs) include military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and commercial automation, inventory management, weather sensing, environmental monitoring, national border monitoring and building and structure monitoring [10]. So, this ensures that possible applications of Wireless Sensor Networks (WSNs) in many scenarios.

Here, we mention some possible applications of Wireless Sensor Networks (WSNs) as shown in Figure 5 below [11, 10]:

❶ **Health Applications**[12, 13, 14] :
In Health applications area, Wireless Sensor Networks (WSNs) provide the interfaces for continued monitoring of physiological patient data, abnormal, drug monitoring in

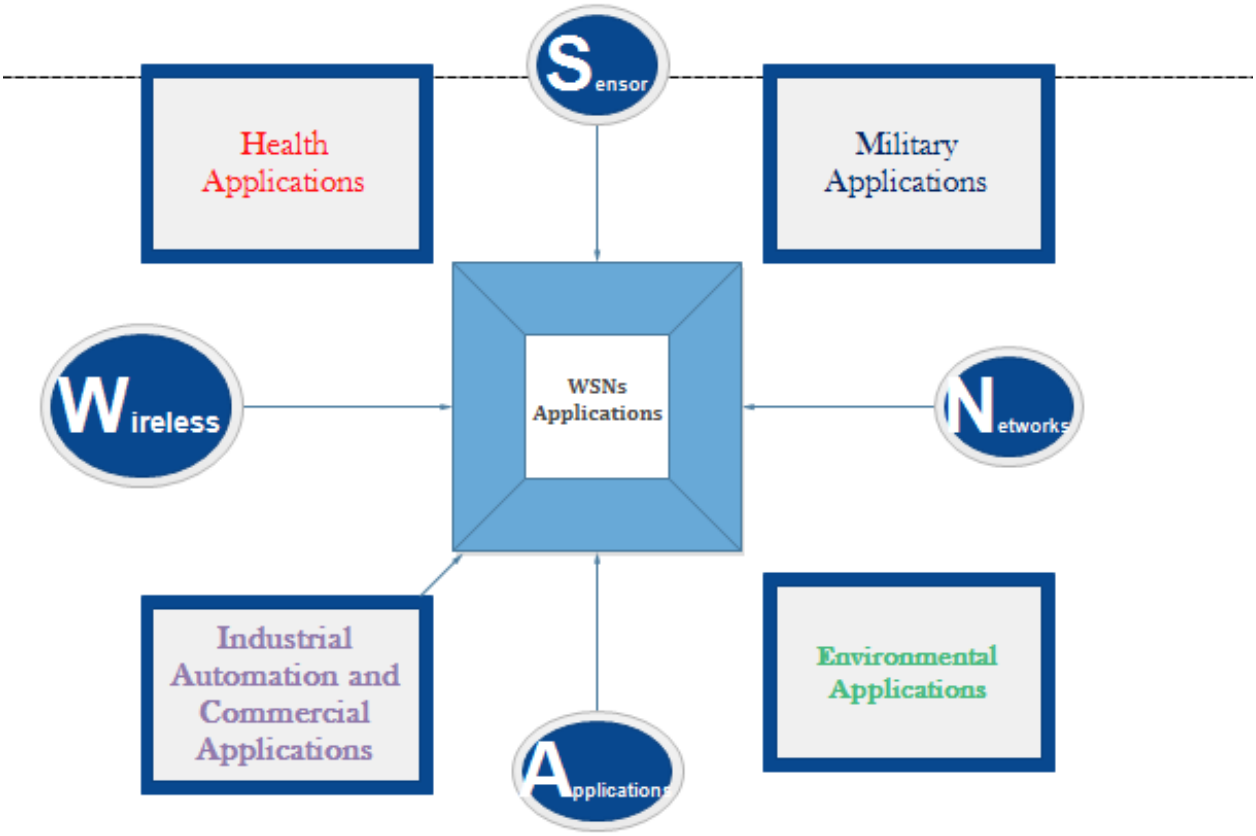

Figure 5. Applications of Wireless Sensor Networks (WSNs)

hospital or clinic, emergency communication channels,diagnostics ,integrated patient monitoring for data management and telemedicine supporting for consultations of human physiological data and tracking and monitoring doctors and patients in a hospital.

❷ **Military Applications**[8] :
Some applications of Wireless Sensor Networks (WSNs) in Military applications include:

- Battle damage assessment
- Targeting
- Nuclear, Biological and chemical detection
- Monitoring friendly forces and equipment
- Tactical Communication
- Sniper Detection
- Battlefield resources monitoring

❸ **Environmental Applications**[15, 8] : Some Environmental applications of Sensor Networks include :

- Agricultural intruder detection

- Tracking the movements of birds , small animals and insects
- Monitoring environmental conditions that affect crops and livestock and also irrigation
- Forest fire detection
- Flood detection
- and more

❹ **Industrial Applications** [16] :
Sensor Networks are a key part of the modern industrial environment to increase productivity and competitiveness. Some industrial application of Sensor Networks includes fault detection, measure temperature, vibration, pressure, pollutants, motion and electrical quantities, alert warning about risks of internal and external conditions.

❺ **Commercial Applications** [8, 14, 17] :
Here we recall some possible industrial applications of sensor networks include building virtual keyboards, monitoring material fatigues, managing inventory, monitoring products quality, interactive museums, factory process control and automation, robotic control and guidance in an automatic manufacturing environment, monitoring disaster areas,machine diagnosis; transportation; factory instrumentation; local control of actuators; detecting and monitoring car thefts; vehicle tracking and detection and so on.

❻ **Home Applications** [18] :
The advances of intelligent sensing agent and sensor-based information appliances will spread pervasive technology that can be integrated with the emerging infrastructure for global information. At all, some possible home applications of sensor nodes are :

- Automatic detection of home/office layouts.
- Automated meter reading Even they allow end users to manage home devices locally and remotely easily.
- Control TV program switches on / off and switch on a usual program.

2.1.2. WSN's enabling Standards

Having standardized technologies in the communication industry is extremely common. Standard protocols enable the technology more attractive for end users and facilitate interoperability between different vendor architectures. Further, openness and large contribution engaged in the development process of standard increase the reliability and safety of the technology. Moreover, many organizations are responsible for improving standard specifications based market needs to fulfill the key requirements they designed for. One way to ensure cooperate of heterogeneous sensors is through pursuing standardization. There are many advanced standards being made Wireless Sensor technology demands

increased in several fields. Particularly, each standard designed for wireless sensor nodes should consider and fulfill the key requirements of Wireless Sensor Networks (WSNs) especially power consumption. In Wireless Networks, these standards define the functions and protocols for sensor nodes required to interface with various types of networks [2, 1]. The common standards in Wireless Networks are IEEE 802.15.4, IEEE 802.15.3, Wibree, ZigBee, WirelessHART 6LoWPAN and more.

2.1.2.1. IEEE 802.15.4

It is the most relevant communication standard for the WSNs. It is designed for wireless sensor applications that require a short range of communication to use energy efficiently and extend node's battery life. The main features of IEEE 802.15.4 are concentrating on low deployment cost, network flexibility, low data rate, low power consumption and low complexity. In Wireless Sensor standards, IEEE 802.15.4 is designed to support protocols of physical and data-link/MAC-sub layers. It is the basic foundation for ZigBee, WirelessHART ,and WiFi standard specifications, to offer complete networking solutions by developing the upper layers which are not addressed in this standard. Fig 6 illustrates the architecture of IEEE 802.15.4 [19]. IEEE 802.15.4 defines two types of network node: Full-Function Device

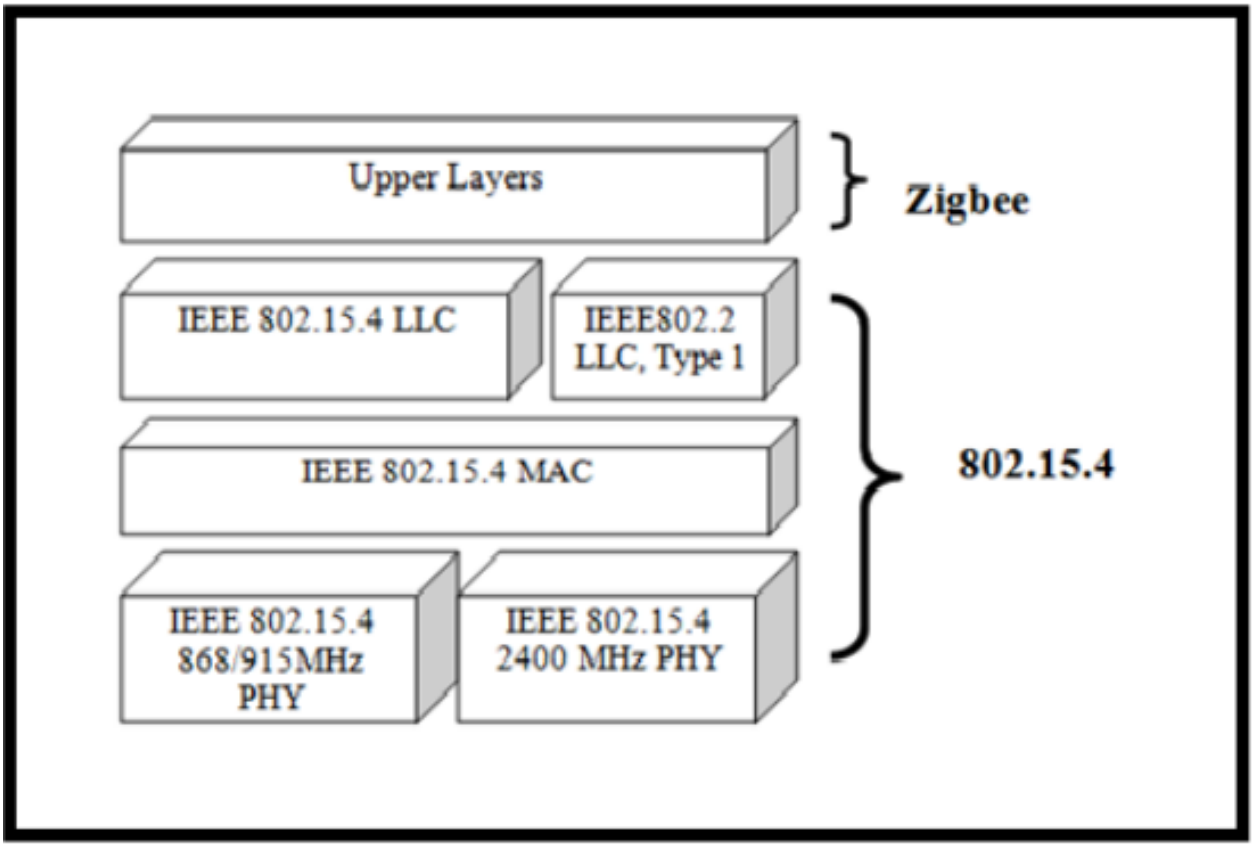

Figure 6. IEEE 802.15.4 Architecture of IEEE 802.15.4

(FFD) and Reduced-Function Devices (RFD) [20]. Full-Function Device (FFD) contains the complete features of IEEE 802.15.4 and serve as the coordinator and as an end device of personal area networks. On the other hand, Reduced-Function Devices (RFD) represents

a simple device with resources and communication requirements. As general, RFD can only communicate with FFD and cannot act as coordinator.In IEEE 802.15.4, the possible interconnections of networks can be built as peer-to-peer or star topology as illustrated in Figure 7. In peer-to-peer model, FFD can communicate with all other devices within its transmission range while an RFD can communicate only with FFD which is currently associated with. A peer-to-peer network can be ad-hoc and self-organizing.

In star model, network nodes (*FFD* and *RFD*) are interconnected in the form of a star and communicate to each other through a central controller.

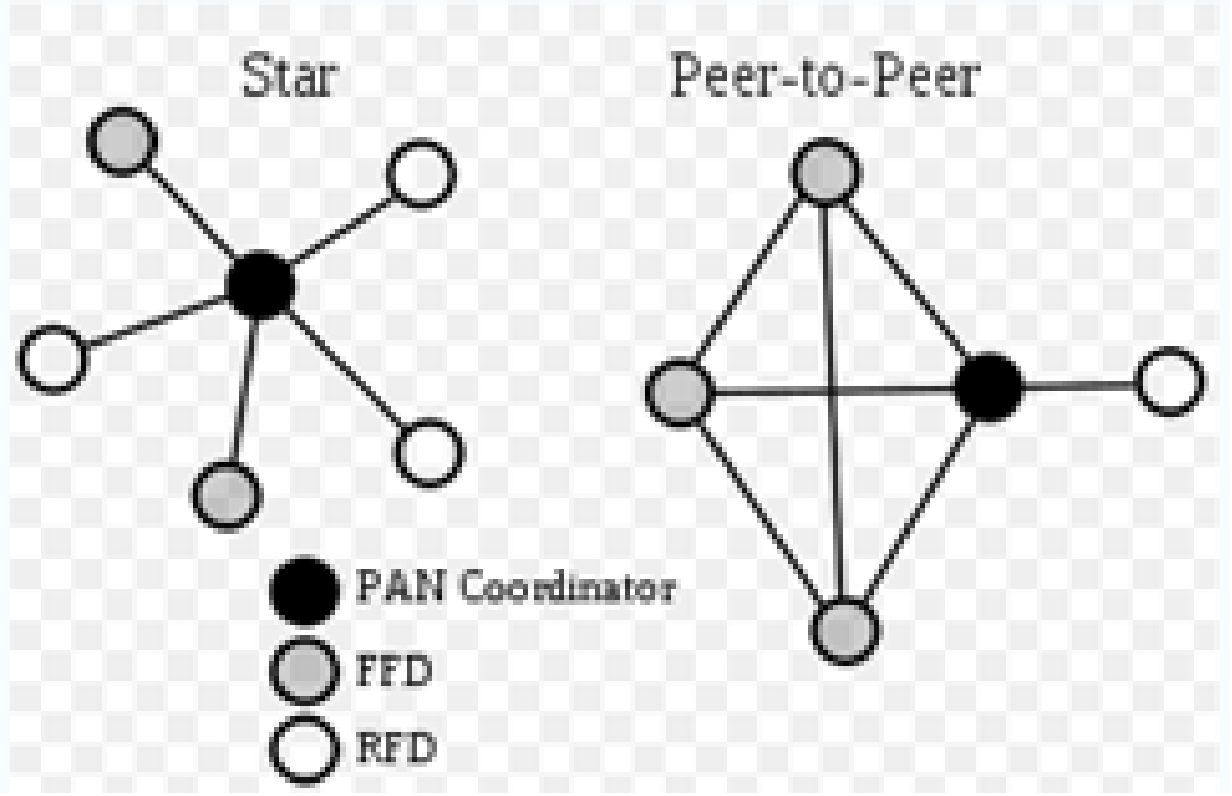

Figure 7. IEEE 802.15.4 Network topologies

2.1.2.2. ZigBee [1, 2]

ZigBee standard has been developed by the ZigBee Alliance. It was designed to address the market interest for being cost-effective, standards-based wireless networking solution that supports low data rates, security, low power consumption and reliability through Wireless Personal Area Networks (WPANs) [21]. It is expected to provide low cost and low power connectivity for a device that requires battery life as long as for several months but doesn't require high rate data transfers. ZigBee defines the higher layer communication protocols built on the IEEE 8902.15.4 standards for Low rate wireless personal area networks (LRWPAN's).This mean network layer and application layer [20]. In ZigBee network, network layer performs securing and routing frames, discovering and maintaining routes, discovering and storing one-hop neighbors and joining and leaving

the network. An application layer of the ZigBee network consists of application support sub-layer(APS), ZigBee device objects(ZDO) and manufacturer-defined application objects. The APS maintains the binding tables, which enables the matching of two devices based on their services and their needs [20]. ZDO maintains the role of devices(coordinator or end device), discover devices and determine their services. It can allow mesh network topology to connect hundreds to thousands of sensor devices together. Compared to other standards, ZigBee standards use very little power and can operate on cell battery for many years.

The ZigBee standard identifies three types of node that might be present in a ZigBee Networks: ZigBee Coordinator, ZigBee Router ,and ZigBee End devices [21, 19]. The coordinator is responsible for configuring key network parameters, network start and other node admissions and network address assignment. ZigBee uses a *Distributed Address Assignment Mechanism* to allocate network addresses for the joined nodes. Only IEEE 802.15.4 FFD can act as coordinator. Router to act as a relay node to passes message from ZigBee end devices to Router or to the ZigBee coordinator. A ZigBee network router is used to extend network coverage area and increase reliability. End device is the endpoint of ZigBee network can talk to the only coordinator or router. It cannot act as a router to relay data from one device to other which allows the node to sleep a significant amount of the time, thereby; giving long battery life. ZigBee End device corresponds to RFD in IEEE 802.15.4 standard. Wireless Sensor Networks based on ZigBee standard is built on two aspects of IEEE 802.15.4 standard and ZigBee protocol. The ZigBee protocol stack of wireless sensor network consists of four layers which are the physical layer, MAC layer, network layer ,and application layer [19]. 802.15.4 Standard deals with physical and MAC layer while network layer and application layer come under the ZigBee protocol. Fig 8 illustrates the protocol stack of ZigBee [19]. The Physical layer is responsible for

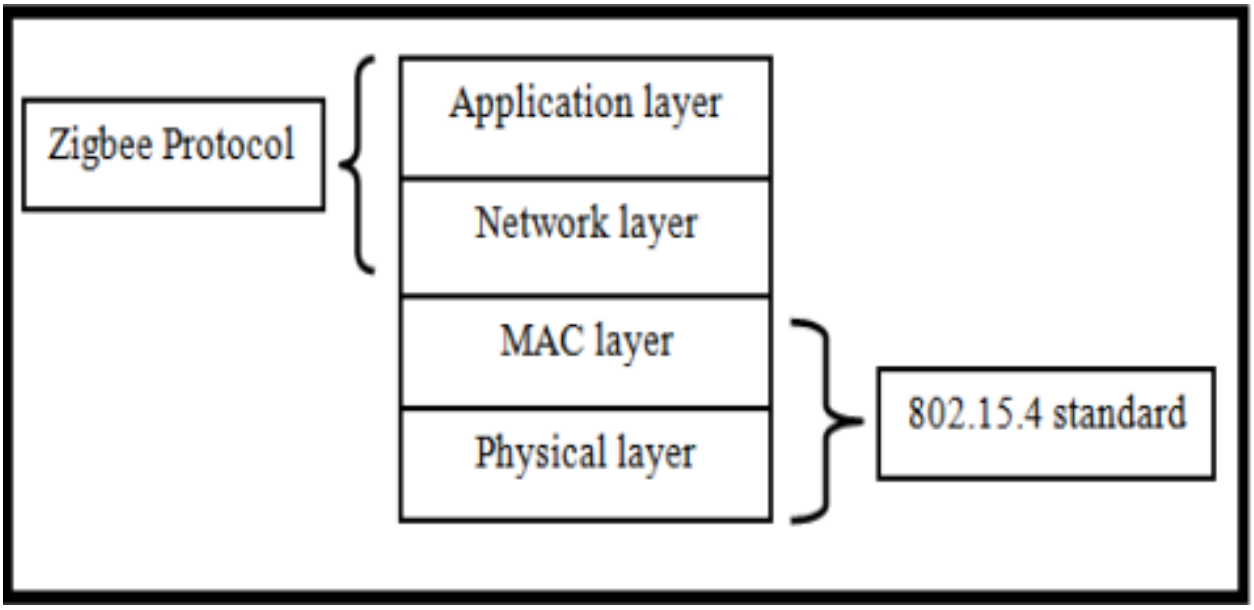

Figure 8. ZigBee Protocol Stack

transmitting and receiving bits, activation, and deactivation of a radio transceiver and energy and signal management functions. Network topologies supported in ZigBee networks are Star, Cluster Tree Topology ,and Mesh topology. In Star topology, using a single PAN coordinator, each device connected directly to the coordinator. In a Star ZigBee network, a coordinator monitors all inter-node communications. A cluster Tree network comprises the collection of Star networks connected whose central nodes are in direct communication with a single PAN Coordinator as shown in Figure 9. The ZigBee Cluster Tree network is formed into an interconnected mesh of Routers and end devices using a set of Routers and a single PAN coordinator. The ZigBee protocol components have the ability to support Mesh Networking. In mesh networking, each node interconnected with all other nodes so that multiple pathways connect to the node as shown in Figure 9. The connections between nodes are dynamically updated and optimized through a built-in mesh routing table. Mesh Networking provides greater stability in failure single node or changing conditions.

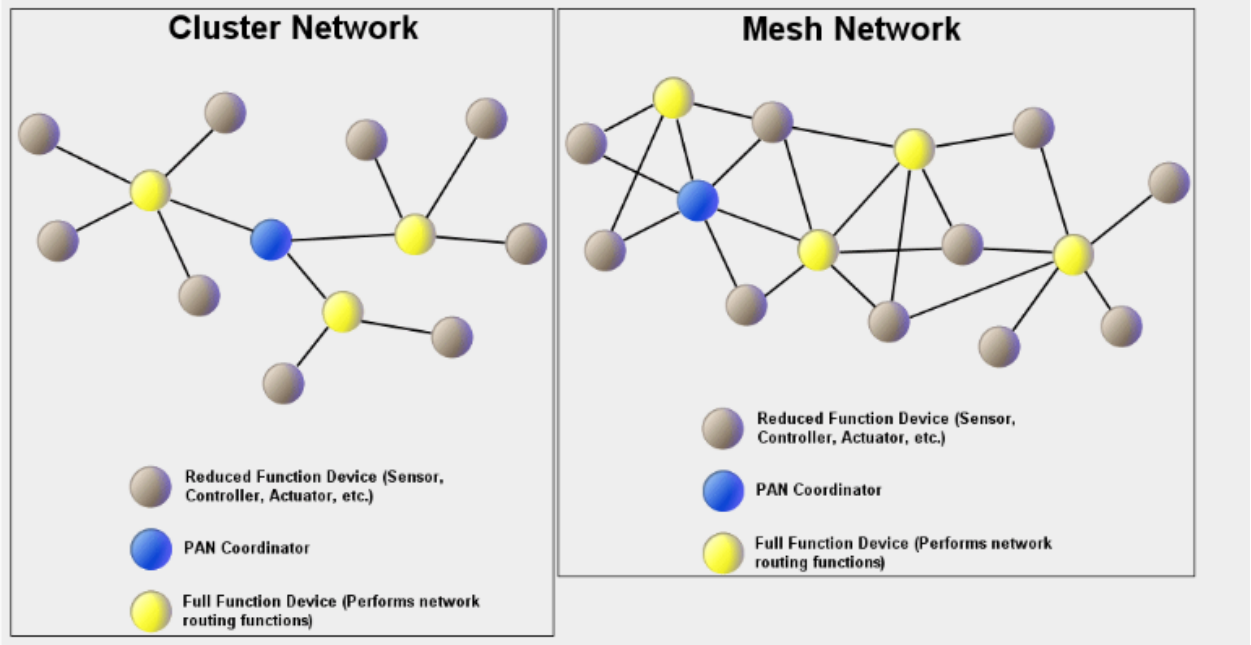

Figure 9. ZigBee Cluster Tree and Mesh Network Topology

2.1.2.3. WirelessHART [1]

It is designed to provide a wireless communication protocol for process measurement and control applications. It relies on the IEEE 802.15.4 PHY layer standard for the 2.4 GHz band operation. It has been developed as a wireless extension to the standard HART, which is used in automation and industrial applications as communication protocol [22]. HART offer master/slaves communication scheme where up to two masters are accommodated in the network.

2.1.2.4. 6LoWPAN [1, 2]

The Internet Engineering Task Force (IETF) develops IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) standard to integrate WSNs with the Internet. It defines the implementation of IPv6 on the top IEEE 802.15.4 to allow any sensor devices to accessible from and with the Internet.

2.1.3. Enabling Technologies

2.1.3.1. Sensor Nodes

The sensor node is a special device envisioned that the intrinsic eyes of Wireless Sensor Networks (WSNs). They are the main component of Wireless Sensor Networks (WSNs) which enable the reason for the deployment of Sensor Networks.

2.1.3.2. Operating System

In addition to Hardware platforms, several software platforms have been developed specifically for Wireless Sensor Networks (WSNs). Among these, the most important platform is an operating system which is designed to support node operation. Traditional operating System manages computing resources, control peripheral devices and provide software abstraction into the applications and therefore they designed to manage processes, memory, file system, devices and CPU time. Traditional Operating Systems are usually implemented in the module and layered fashion including a higher layer of system libraries and a lower layer of kernels. Thus, traditional Operating Systems are not desired for Wireless Sensor Networks (WSNs) because WSNs have constrained resources and a variable topology. Therefore WSNs require a new Operating System that can satisfy their underlined constraint resources, network behavior, and data-centric application requirements. Among the emerging Operating System solution for the sensor node, TinyOS [23] is open source Operating System designed for Wireless Embedded Sensor Networks [1] and allows the applications to directly access hardware when required. TinyOS is WSNs that attempts to address how to guarantee concurrent data flow between sensor devices and how to provide modularized components with less processing and storage overhead[23]. It's component-based architecture, which minimizes the code size and provides flexible implementing new protocols. Its component library includes sensor drivers, distributed services, data acquisition tools and network protocols. TinyOS uses an event-driven execution model for power management strategies and to support high levels of concurrent application in a very small memory size. TinyOS is a power-aware operating system. It can rapidly create tasks associated with events with no blocking and polling and enable the process maintained to sleep to conserve energy when the CPU is idle. TinyOS uses the scheduler to schedule the operation of its component tasks. After performing their function, software components are going to sleep mode. If any data arrived, the event signals to the right component.

Software components also request a task scheduler to perform tasks. In addition to TinyOS, other alternative Operating System solution for sensor nodes includes Contiki [24], a lightweight memory efficient open-source Operating System for tiny networked sensors. Contiki offers several standards of Operating System features such as timers, random number generators, clock, proto-threads and a file system support. It has the IPv6 stack with support for UDP and TCP connections. SOS [25] designed to improve TinyOS by providing dynamic memory allocation, a loadable module, and kernels. MANTIS [26], SensorOS[27] are a multithreading operating systems that support pre-emptive thread scheduling. A pre-emptive scheduler is particularly important to enforce fairness among threads or tasks.

2.1.3.3. Communication Technology

There are several types technologies adopted as communication technology used in Wireless Sensor Networks (WSNs).

One of enabling communication technology promising for WSNs is IEEE 802.15.1/Bluetooth technology. It was standardized by IEEE as Wireless Personal Area Networks (WPANs) [28]. Bluetooth is infrastructure-less short-range RF technology intended to facilitate the communication between electronic devices and with the internet. This technology requires adhoc configuration of master/slave piconets. It supports spontaneous connections between devices to transfer data between units over distances of nominally up to 10 m. Bluetooth can transfer data up to 1 Mbps. Supported devices include laptops, PCs, cell phones, mobile devices and so on. However, the application involving a large number of tiny devices using adhoc networking will face a number of challenges when using Bluetooth as communication technology. Another interested communication technology is ZigBee [22]. It is a specification for a suite of high-level communication protocols layers using low power digital radios and builds upon Physical Layer (PHY) and Medium Access Control (MAC) layers in IEEE 802.15.4 specification. ZigBee enables peer-to-peer communication at higher level protocol. The accepted frequency bands in the ZigBee Wireless Network are 868/915 MHz and 2.45 GHz and communication range is 75 meters. The communication can only be between one master and 255 slave nodes(maximum number of nodes in ZigBee networks). Further explanation of about the ZigBee specification is illustrated in section 2.1.2.2.

2.1.4. WSN Architectures

A Wireless Sensor Network consists of a large number of nodes that are collaborating to achieve a sensing task in an application deployment area. Due to energy power constraints of nodes, network design has an effect on the energy consumption, which impacts the operational lifetime of WSNs. Therefore, WSNs require network protocols to implement several management functions and network control, for instance, node localization, network

security, data aggregation, synchronization, and medium access control. However, network protocols of tradition wireless networks cannot be applied directly to WSNs, because they do not consider storage, computation and energy constraints in sensor nodes. Moreover, WSNs are application specific. Because of these, a Wireless Sensor Network requires a new network protocol which considers resource constraints in sensor nodes as well as requirements of different WSNs applications.

2.1.4.1. Sensor Node Structure

Sensor Nodes are made up four main hardware components as shown in Fig 10 [29]: A Sensor Unit, Processing unit, Power unit, and a transceiver unit. They also have other application-dependent components like location finding system, power generator, a mobilizer and so on. Sensor units are usually a bunch of two sub-units: Sensor and Analog to Digital Converters (ADCs). A sensor is WSNs node which can obtain environmental phenomenon status. A signal obtained by the sensor nodes in charge of collecting and observing environmental is converting to into digital using ADCs and transferring into the processing unit. The processing unit receives data from sensor unit and processes accordingly. The processing unit, which is associated with a small storage unit of sensor nodes, manages the procedures that sensor nodes are collaborating to carry out sensing tasks. A transceiver unit then enables the nodes connected to the network. And it also transfers data to realize the achievements of physical communication of sensor nodes. Power unit provides reliable power energy for sensor nodes to enabling Sensor Networks technologies. Wireless Sensor Networks (WSNs) may use other application-dependent components to optimize the assigned sensing task procedures. A number of sensor node routing techniques, sensing tasks and node localization require a knowledge of the location with high accuracy. For that reason, a sensor node may require location finder in some application to localize each sensor node. In some cases, it is assumed that some of the sensor nodes equipped with Global Positioning Systems (GPS) unit. In addition to this, WSNs may use additional techniques for localization techniques: centroid, APIT, Spot-Light, Radio Interferometric Geolocation [13]. A mobilizer is an option part needed to move sensor nodes when it's required to carry out assigned tasks.

2.1.4.2. Network architecture

A Wireless Sensor Network is the collaboration of sensor nodes deployed in an area of sensing and one or more BS located far away, close to or inside the sensing area, as shown in Figure 11. Once sensor nodes deployed in sensing region, BS sends commands into nodes while they collaborate to accomplish sensing tasks and send sensed data back to the BS. Meanwhile, BS collects sensed data from the collection of sensor nodes and forwards the processed data over the internet(sometimes satellite) to the users who requested it.

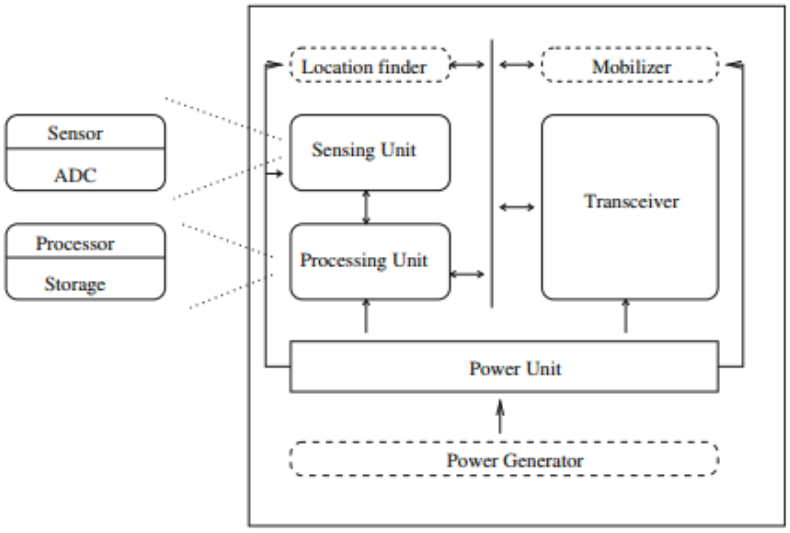

Figure 10. Sensor Node Structure

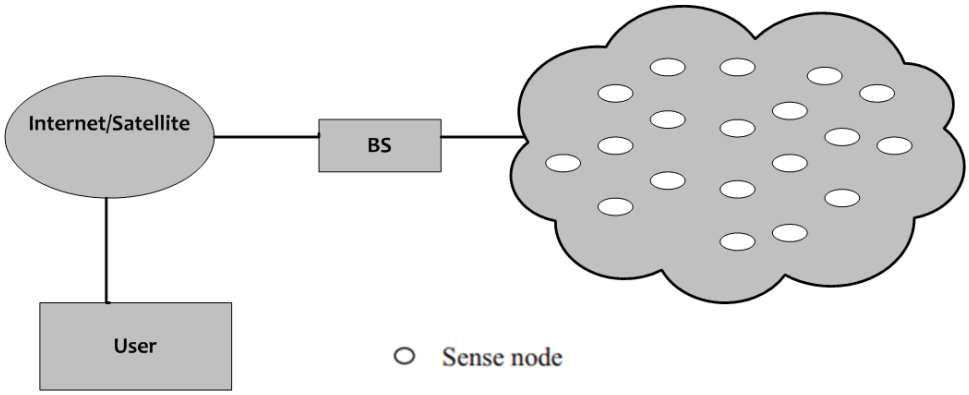

Figure 11. Sensor Network architectures

The network architecture of WSNs can be limited two architectures: Single-hop network architecture and multi-hop architecture. In single-hop network architecture, each sensor node uses a single-hop to transmit sensed data back to BS as shown in Figure 12. However, this network architecture is costly in terms of energy consumption over long distance transmission.In many cases, energy consumed for communication is much higher than that for sensing and computation in WSNs. Therefore, it is important to reduce the number of packets and transmission distance in order to save energy consumption and extending network lifespan. Compared to Single-hop network architecture, the multi-hop network architecture is highly preferred in case of energy consumption and network lifetime. In multihop network architecture, sensor nodes are deployed densely and close to each other to reduce long-distance transmission. In multihop communication, the sensor node transmits sensed data to BS using one or more nodes as a relay. In other words, they use short-distance for sensed data transmission, which can reduce energy consumption for communication. Multihop network architecture can be organized into two types: Distributed and Hierarchical [30].

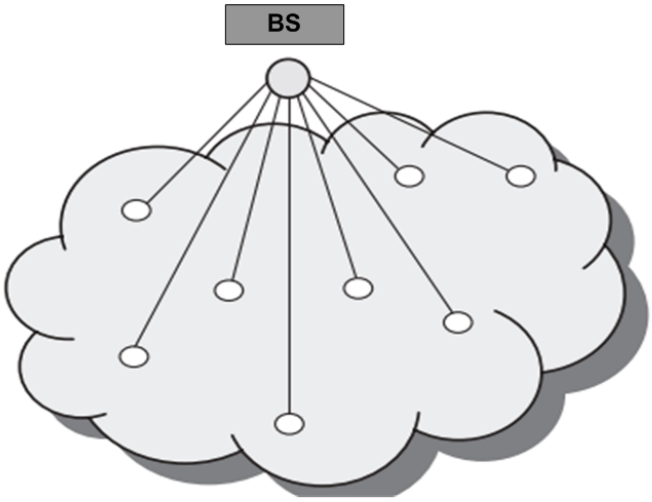

Figure 12. Single-hop Network architecture

❶ **Distribute architecture**: In a distributed architecture, each sensor node has the same role and all nodes are peers. In this architecture, most of the time data collection is accomplished using data-centric routing technique, in which BS transmits a command to all sensor nodes and only the nodes that have the matching data with the query will respond to the BS. Each node uses a multihop to route data to the BS and uses its neighbor nodes as a relay as illustrated in figure 13.

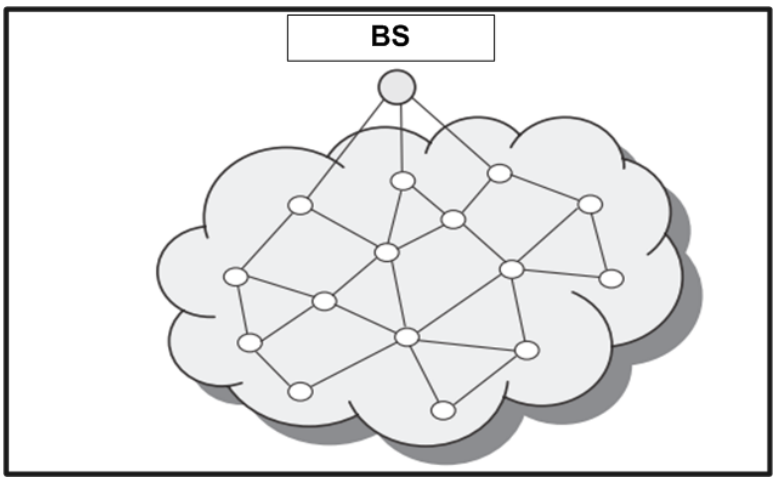

Figure 13. Flat Network Architecture

❷ **Hierarchical Architecture** : In this architecture, sensor nodes are divided into clusters, where the CH act as a local aggregator to collect data sensed by cluster members and forwards to the BS. In other words, CH serves as relays for transmitting the sensor node data to the BS as shown in figure 14. A node with high energy can be elected as Cluster Head (CH) to process and transmit data to the BS while a node with low energy can be used for sensing tasks and sensed data to its Cluster Head.Beside to reduce energy consumption for communication, this architecture has many advantages

when compared with distributed architecture:

- Improve scalability and balance traffic load in large networks
- Local data aggregation in order to reduce the amount of data transmitted to the BS .

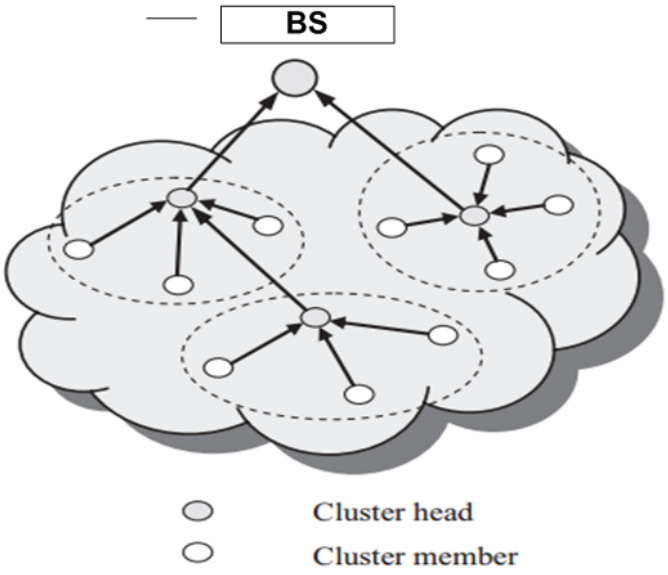

Figure 14. Single-hop clustering Architecture

2.1.4.3. WSN Protocol Stacks

The WSNs protocol stack consists of five protocol layers [1]: the physical, network, transport and application layers as shown in figure 15.

❶ **Application layer**: This layer consists of several application layer protocols that perform various WSNs applications, for example, network security, node localization, data dissemination and time synchronization. Some of these protocols are SMP, SQTL ,and SQDDP. For instance, Sensor Management Protocol (SMP) is an application layer management protocol that provides application operations to carry out tasks such as synchronizing nodes, sensor node's mobility, scheduling sensor nodes, querying the sensor node status and data related sensor node locations. The sensor query and data dissemination protocol (SQDDP) is application layer protocol concerns with WSNs queries. It provides user application with the interface to send queries, respond to queries and collect responses [8]. The sensor query and tasking language (SQTL) is known as sensor programming language which used to implement middleware in Wireless Sensor Networks [31].

❷ **Transport Layer**: It is responsible for reliable end-to-end data delivery between the sensor node and BS. In other words, responsible for reliable data delivery required by the application layer. Due to Sensor node constraints, such as energy, storage, and

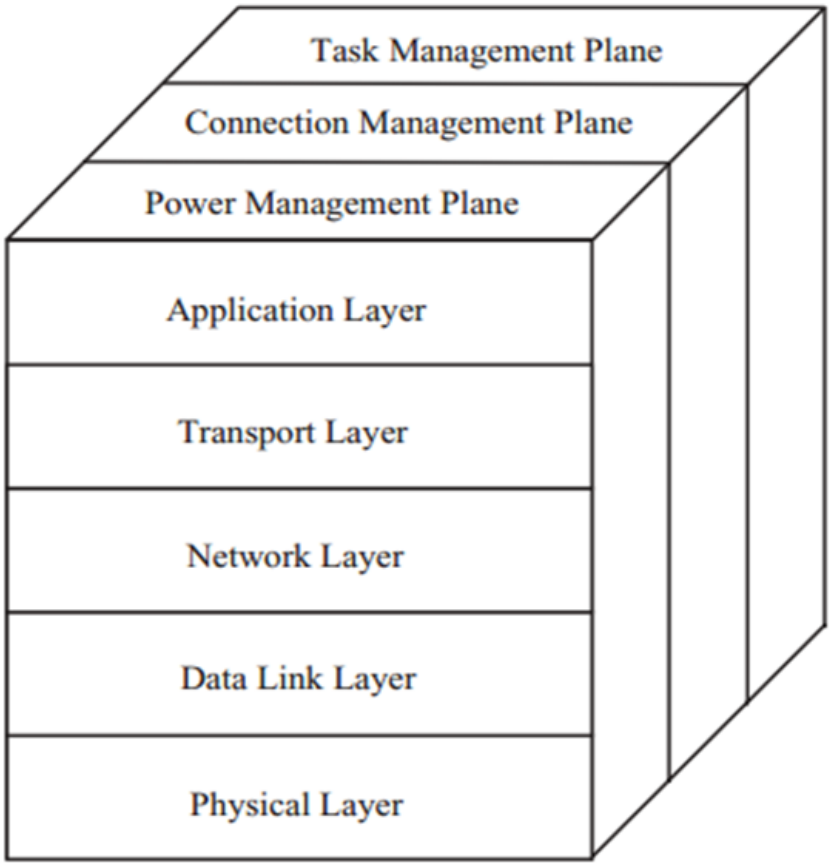

Figure 15. WSNs protocol stacks

computation constraints, we cannot apply traditional transport protocol directly to WSNs. For example, retransmission and window-based congestion problem used in TCP cannot be used directly for WSNs since they are not efficient in resource utilization. In WSNs, transport layer protocol is used for two main functionalities: congestion control and end-to-end reliability. However, factors such as energy consumption and limited resources, prevent end to end reliability being employed in WSNs. Data delivery in WSNs occurs in two directions: Upstream and downstream. In the upstream approach, the sensor node transmits their sensed data to the BS, while in downstream approach the data sources such as commands, queries, are originated from BS to Sensor nodes, which require a highly reliable delivery ratio. In both directions, the data flow may have different reliability requirements. For example, in upstream, to some extent, the sensed data may be correlated or redundant so that data flows in this approach is less tolerant. On the other hand, the data flows in downstream, such as queries, commands and programming binary may require 100‰reliable delivery. Therefore, WSNs require local reliability mechanisms.

❸ **Network layer**: It is responsible for routing the sensed data from sensor nodes to the BS. In WSNs, sensor nodes are deployed in sensing area to observe phenomena. The observed data need to be transmitted to the BS via single-hop long-range communica-

tion or multihop short-range wireless communication. However,single-hop long-range communication is not suitable in many cases because of resource-limited in WSNs. In contrast, multihop short-range communication is preferable than single-hop in two cases: reduce the energy consumption of sensor nodes and reduce the signal propagation and channel fading effects inherent in long-range wireless communication effectively. In densely deployed sensor networks, sensor nodes and neighbor nodes are close to each other, thus, it is possible to use multihop short-range communication. In multihop communication, to send sensed data to the BS a sensor node should employ a routing protocol to select an energy efficient multihop path from the source node to the BS. In traditional routing protocol do not consider energy consumption so that we cannot apply this routing protocol directly to WSNs. Therefore, network layer should be designed with energy efficiency consideration of sensor nodes.

❹ **Data-link layer**: It is responsible for data multiplexing, data frame creation, medium access control and error control to provide point -to- point and point-to-multipoint transmissions. Specifically, the important function of the data-link layer is Medium Access Control (MAC), which facilitates shared communication resources or medium among various of sensor nodes. However, MAC applied to WSNs requires a modification for the reason of the unique characteristics of sensor nodes, specifically, the energy resource constraint. The primary concern of WSNs is energy saving for extending network lifespan, due to that traditional MAC cannot be suitable for WSNs.

❺ **Physical layer**: The bit streams forwarded from data-link layer require conversion to signal in order to be transmitted over a communication medium. For this purpose, the physical layer deals with this conversion.

Moreover, the protocol stack can be divided into a group of management planes, including power, connection ,and task management planes. The power management plane is responsible for monitoring the efficient use of the power level of a sensor node for sensing, processing, and transmission and reception by implementing efficiency power management mechanisms at different layers. For example, at the MAC layer, a sensor node can sleep when there is no data to transmit by the turn of its transceiver. At the network layer, a sensor node may select a neighbor node with high residual energy as its next hop to the BS or using the most energy-efficiency routing protocol. At the transport layer, probing can alter Transport Control Protocols (TCP) retransmission operations by reducing unnecessary retransmission, thus achieving power conservation and higher throughput. The connection management is responsible for sensor node configuration and reconfiguration to establish and maintain networks in case of node deployment and network partitions/topology change due to node mobility, failure, clustering, node addition and so on. The task management is responsible for managing and monitoring task distribution among sensor nodes in a sensing area in order to assure energy efficiency and extending network lifetime.

2.1.5. WSNs Classifications

A Wireless Sensor Network is usually application specific and thus has different characteristics. Based on different criteria, WSNs can be organized into different categories:

❶ **Mobile and Static Network**: This categorization is based on the node movement. Based on sensor node mobility, WSNs can be static or mobile. In static WSNs, all sensor nodes are stationary without movement. However, in mobile WSNs, all sensor nodes are mobile, where sensor node moves one location to another to sense data and thus frequently network topology changes.

❷ **Deterministic vs Non-deterministic**: Depending on sensor node deployment, WSNs can be deterministic or non-deterministic. In deterministic WSNs, sensor node's positions are preplanned and fixed once deployed. Instead, in non-deterministic, sensor nodes are deployed randomly without preplanning.

❸ **Single-hop vs Multihop Network**: According to the number of hops between node and BS, WSNs can be a single-hop or multihop network. In single-hop WSNs, all sensor nodes use single hop to transmit the sensed data to the BS while in multi-hop WSNs, sensor node uses multi-hop communication to transmit sensed data to the BS.

❹ **Homogeneous vs Heterogeneous**: Depending on sensor node capabilities, a WSNs can be homogeneous or heterogeneous. In homogeneous WSNs, all sensor nodes have the same capability in terms computation, energy ,and storage. In contrast, a heterogeneous WSNs has additional known as actuator nodes with high capabilities in terms of processing and communicating than normal sensor nodes.

2.1.6. Node Clustering for WSN

Partitioning Sensor nodes into clusters has gained increasing importance in several research communities in order to save energy consumption and prolongs the network lifetime. Every cluster holds local coordinator of transmission, often referred to as the Cluster Head (CH). In many cases, a CH has been elected among sensor nodes. Although many clustering algorithms have been proposed by the research community for Wireless Sensor Networks. Many clustering strategies have been proposed for WSNs using different parameters [32]. These strategies contribute several achievements for WSNs: extending the WSNs lifespan, efficiently using energy, reducing communication cost.

Furthermore, at various clustering schemes are introduced in section 2.1.6.1. A comparison in terms of their performance in CH selection, network lifetime, fair distribution of CH, communication cost, energy consumption is also carried out between the schemes. Finally, several topology models of WSNs are introduced in section 2.1.6.2.

2.1.6.1. Node Clustering Schemes

Node clustering is an effective mean for managing a large number of sensor nodes in WSNs. As a whole, node clustering algorithms operate in two phases:

- Clustering Set-up: This phase includes a CH selection between sensor nodes and other nodes joining the selected CH forms clusters.
- Clustering maintenance: Once cluster established, the configuration of the cluster may be changed due to node dies, movements or network topology changes.

Several node clustering schemes have been proposed to achieve the objectives of WSNs and some of them are:

- **LEACH**
  A LEACH [33] is a hierarchical clustering scheme for WSNs, which forms clusters based on the received signal strength [34] and uses CH as a local gateway to route sensing data back to the BS. The LEACH breaks the WSNs operations into two rounds: setup state and steady-state stages. The set-up stage made-up of cluster formation and CH selection. The steady-state stage represents a period of data transmissions between sensor nodes and Base Station i.e. transmission of sensed data to the Cluster Head (CH) and then to the BS.
  In LEACH, setup state is the stage where each sensor node to decides become either Cluster Head or Cluster Member. To do so, each node chooses a random number between **0** and **1**, which is the probability it will become a CH or not, then compares this number with $\mathcal{T}(v)$ and then v node will become a Cluster Head for the current round r if the chosen number by v is less than $\mathcal{T}(v)$. This $\mathcal{T}(v)$ computed as follows:

$$\mathcal{T}(v) = \begin{cases} \dfrac{\mathcal{P}}{1 - \mathcal{P}\left[r \mod \left(\frac{1}{\mathcal{P}}\right)\right]}, & \text{if node } v \in \mathbb{G} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (1)$$

  Where $\mathcal{P} = \frac{\lambda}{\mathcal{N}}$; λ is the assumed number of CHs in the round and $\mathcal{N}$ is the number of v nodes in the network, r is the current round number, and $\mathbb{G}$ is the set of sensor nodes that have not been CH in $\left(\boldsymbol{r} \mod \left(\frac{\mathcal{N}}{\lambda}\right)\right)$ rounds.
  Therefore, from the equation (1) we can say that, for the initial round $r\left(r = 0 and r = \frac{\mathcal{N}}{\lambda}\right)$, each node has the same probability $\mathcal{P}$ to become a CH. However, when r increases the $\mathcal{T}(v) = \mathbf{0}$ for nodes that have been a CH. For example, for rounds, $r = \mathbf{1} \rightarrow \frac{N}{\lambda} - 1$, the v nodes that have been a CH have a value of $\mathcal{T}(v) = \mathbf{0}$. In other words, they cannot become a CH. In contrast to this, $\mathcal{T}(v)$ increases as r increases for v nodes that have not been a CH.

Moreover, $\mathcal{T}(v)$ value is increased as round r number is increased in the next round while desired CH candidates have been reduced. In this case, the nodes v elected as a CH broadcasts to all sensor nodes in the network, advertising its cluster head status are the newest CH. The sensor nodes that have not been a CH in the current round to the join closest cluster based on the signal strength received from new elected CH. Each sensor node sends join request to the CH and cluster formations are organized.

However, LEACH uses single-hop long communication so that sensor nodes transmit sensed data directly to CH and then, CH transmits gathered data back to the BS directly, regardless of the distance between a CH and the BS. Due to energy constraints, this technique is not effective in large-scale WSNs when BS is far away from the deployment area of sensor nodes. The CHs elected randomly without the consideration of energy level so that there is a chance a node with a low energy level gets elected as a CH, which can dysfunctional of the network. Consequently, it is not applicable for networks deployed on a large scale.

- **T-LEACH**

  The T-LEACH protocol has been proposed in [35] as an improvement over LEACH. Instead of a probabilistic decision, T-LEACH replaces CH based on a threshold value of residual energy on the sensor nodes. It minimizes the cluster head selection overhead by using a threshold of residual energy. However, T-LEACH uses a random selection process like LEACH to elect a CH regardless of specifying any criteria how CH has been elected.

- **PEGASIS**

  In [5], PEGASIS Clustering algorithm has been proposed to show improvement over LEACH protocol. In this protocol, instead of multiple of clusters, every sensor node form chains by communicating only with adjacent neighbor nodes by adjusting its power signal to be only heard by this adjacent neighbor node. Each sensor node uses signal strength to estimate the distance to neighborhood nodes in locating the closest nodes. On the basis of energy residual of the nodes, one node is selected from the chain formed to transmit the data collected from its adjacent nodes to the BS.

  Unlike LEACH, PEGASIS form chains which consequently avoids cluster formation. But it uses only one node in a chain to transmit the collected data back to the BS instead using multiple nodes as in the case of protocol in [33]. Consequently, the leader can become a bottleneck with traffic overhead of the whole chain.

- **I-LEACH**

  I-LEACH is proposed in [36] in order to overcome the two shortcomings existing in LEACH counterpart. In I-LEACH, the probability based election

criteria of a CH in LEACH was replaced with the energy level concept , which is a significant improvement over LEACH. Besides, It uses $x-axis$ coordinate of the nodes to form clusters ; it's used in the uniform distribution of the CHs, which remain a CH close to every sensor node. Therefore, non-CH will not require transmitting their data over long distance. I-LEACH showed substantial improvement over LEACH. However, still it was not as much efficient as compared to PEGASIS in terms of a number of completed rounds for a given amount of energy.

- **U-LEACH**
  U-LEACH [37], combines the features of PEGASIS and I-LEACH as an improvement over LEACH. Like I-LEACH, U-LEACH forms clusters based on the $x-axis$ coordinate values of the nodes. The CH election process is based on the residual energy of sensor nodes. U-LEACH uses multihop-short distance transmission approach to transmitting data from remote nodes to CHs and then from CHs to other CHs. Finally, the collected data delivered to the BS in multihop away fashion.

2.1.6.2. WSN topologies

Topology is one requirement should be considered in Wireless Sensor Networks (WSNs) designs while the number of sensor nodes is huge in the network. The most common network topologies considered in WSNs are a single-hop flat model, single-hop clustering model, multihop flat model and multi-hop clustering model [38].

In the single-hop models, all sensor nodes/CH are communicating and transmit their data to the Base Station (BS) directly as shown in Figure 16 and Figure 17. These topology models are applicable only in small-scale environments of Wireless Sensor Networks (WSNs). Because of cost expensive in terms of energy consumption and in the poorest case, the Base Station (BS) may be unreachable. Therefore, these architectures are not feasible in large-scale areas of WSNs and do not balance energy consumption along with the WSNs networks.

Figure 18 and Figure 19 depicts the two multi-hop models respectively. In multi-hop flat model, the sensor nodes can communicate with the Base Station (BS) directly (single-hop flat model) or use another node as a gateway(single-hop clustering model). Particularly, in the multi-hop way of this model, all nodes should share the same information like routing table, due to that communication overhead and energy consumption can be proliferated. Moreover, the communication channel resource is shared and managed by a single node which results in low efficiency in the resource usage. On the other side, multi-hop clustering model is designed with minimum with aforesaid limitations of the WSNs topology models. In this model, data aggregation and transmission has been performed by Cluster Head (CH). In other words, Cluster Head (CH) acts as a gateway between end nodes and Base Station

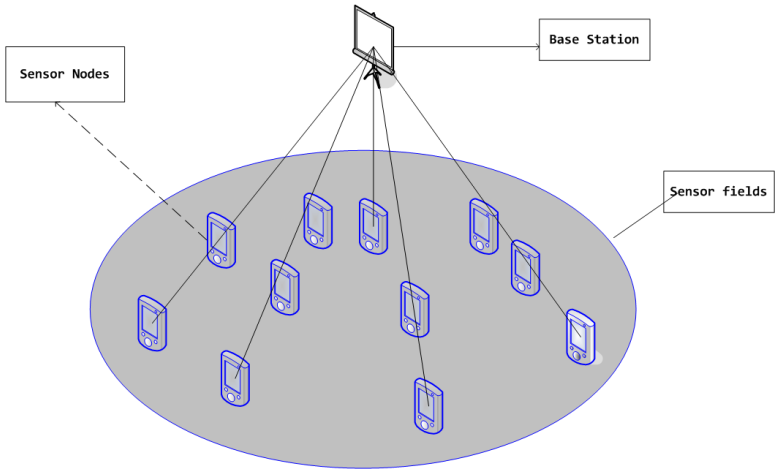

Figure 16. Single-Hop flat Model

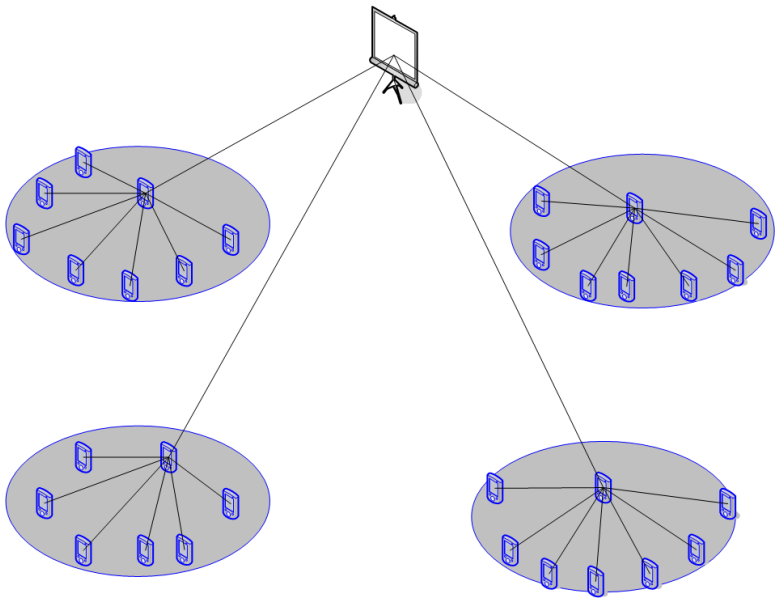

Figure 17. Single-Hop Clustering Model

to transmit and for communication purpose. Because of that this model can maintain and optimize low energy consumption, communication overhead and extend the network communication range. Additionally, this model can preserve collision reduction between Clusters and are reuse cluster by cluster through orthogonal medium resource allocation by each cluster. In other words, data transmission and operations are scheduled which requires time synchronization between all sensor nodes. Therefore, multi-hop clustering model is feasible in large-scale areas of Wireless Sensor Networks (WSNs) deployments.

2.1.7. WSN Design Challenges and Vulnarabilities

A Wireless Sensor Networks (WSNs) are expected to be a solution to many applications as discussed in Sec 2.1.1. However, due to the unique characteristics of WSNs, the security

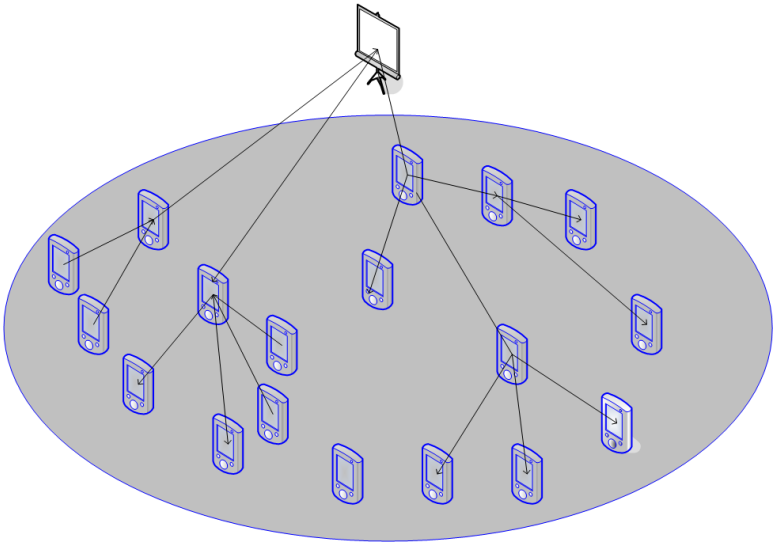

Figure 18. Multi-hop flat model

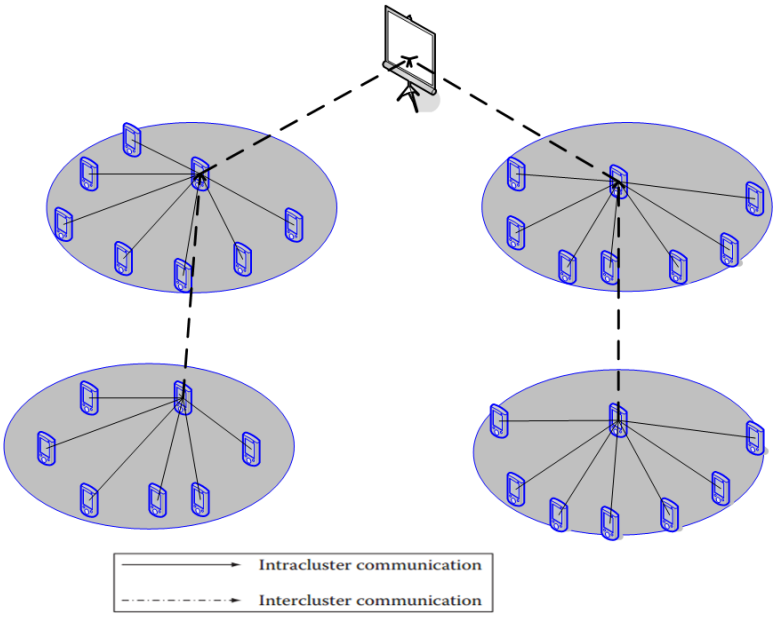

Figure 19. Multi-hop flat model

of this network will be vulnerable to different security issues. Inherently WSNs have many constraints compared to another type of networks which make it difficult to apply directly existing security mechanisms used in conventional networks to wireless sensor networks. Most sensor nodes in WSNs are resources retained in terms of energy, computation, communication capabilities and so on [39]. Thus, each security service designed for WSNs must consider the key constraints of sensor nodes [40].

2.1.7.1. Limited Resources

The design and implementation of security approaches for WSNs have required a certain amount of sensor node's resources, including memory, code space, and energy to switch sensor nodes.However, higher security levels in WSNs usually consume high

energy resources for communication between nodes, message expansion ,and security protocol functions. Thus aforementioned resources are very limited in WSNs. Primarily sensor node uses flash memory to store downloaded application code and RAM for storing application programs, sensor data, and intermediate computations [3, 41]. Therefore, it is very difficult to load complex algorithm after loading the OS and application code. For instance, TelosB(basic sensor type) has 1024 K flash storage, 16 bits,8 MHz RISC CPU with only 10 K RAM and 48K program memory. With such scarcity, the software designed for wireless sensor node should be simple. Beside this, energy power is the most constraint resources in wireless sensor networks. Once sensor nodes deployed in WSNs, it is not possible to replace or recharge the battery of sensor nodes in several scenarios. So when implementing security protocols within sensor node, the influence of added security protocol must be considered. Energy is the most intrinsic resource for WSNs. Communication performed between sensor nodes are require high power energy. Thus the security protocol should provide special effort to be communication efficient in order to be energy efficient [42].

2.1.7.2. Wireless Medium

Wireless Sensor Networks (WSNs) use broadcast mode of communication between sensor nodes to expand the messages over the wireless medium. However, the broadcast nature of wireless medium can make eavesdropping of a message by third parties. An adversary can replay, alter, intercept the transmission going on in the networks. In other words, the attacker can inject malicious code into a valid packet transmitted over the wireless medium. Clearly, the security protocol should prevent the altering, intercept, denial of service or replaying of packets in sensor networks.

2.1.7.3. Unreliable Communication

Unreliable communication is another challenge into wireless sensor security. The security of the network based on heavily on defined protocol, which in turn depends on communication [41]:

- Unreliable Transfer : Normally the packet-based routing of a sensor network is connectionless and thus inherently unreliable. Packets may get adverse effect due to channel errors which result in the missing or lost of packets. In addition, the unreliable wireless communication channel may also consequence in the packets damaging.
- Conflicts : Even if the channel is reliable, the communication may be still unreliable due to the broadcast nature of Wireless Sensor Networks. This problem is common in highly crowded sensor networks. The packet will conflict with each other in the middle of the transfer and may result in unreliable transmission.

- Latency : The multi-hop routing, network congestion and node processing can lead to greater latency in the sensor networks, thus making it difficult to achieve synchronization among sensor nodes.

2.1.7.4. Hostile Environment

It is well known that impossible deployment of traditional networks in the hostile environment. A WSNs is a new technology coming up with a solution to this issue. Wireless Sensor Networks (WSNs) can simply deploy in a hostile environment to control and monitoring application service. However, a hostile environment may face the possible destruction of networks or physically capturing of sensor nodes. Since a node may be in an unprotected area, the attacker can easily capture node physically to gain physical access or compromise to launch internal attacks. Particularly, an attacker can clone physically captured node to compromise Wireless Sensor Networks. So the highly hostile environment is a serious challenge for designing security protocols in WSNs.

2.1.7.5. Unattended Operation

Depending on the function of the particular sensor network, the sensor node may be left unattended for a long period of time [42]. There are three cautions to unattended sensor nodes[41]:

- Exposure to Physical attacks: The sensor node may deploy in an environment open to adversaries, bad weather and so on. The possibility that a sensor suffers a physical attack in such an environment is therefore much higher than the typical PCs, which are located in a secure place and mainly face attacks from a network.
- Managed Remotely : Remote management of a sensor network makes it virtually impossible to detect physical tampering (i.e., through tamper-proof seals) and physical maintenance issues (e.g., battery replacement). Perhaps the most extreme example of this is a sensor node used for remote reconnaissance missions behind enemy lines. In such a case, the node may not have any physical contact with friendly forces once deployed.
- No Central Management Point: A sensor network can be a distributed network without a central management point. This will increase the vitality of the sensor network. However, if designed incorrectly, it will make the network organization difficult, inefficient, and fragile.

2.2. Security Issues in WSNs

A WSNs is a special type of networks requires reliable security approaches. Since WSNs usually used in a confidential and sensitive environment, the accuracy of data and the health of this networks are significantly important. In WSNs, security is a critical issue and vital requirements due to several constraints and challenges. Broadcast and wireless nature of transmission medium, unsafe deployment of nodes and unattended nature of WSNs might make vulnerable against security attacks. The hardware and energy constraints in Wireless Sensor Networks being made the difficulty designing of security protocols in WSNs. Furthermore, this section has reported basic overview concerning security issues in Wireless Sensor Networks (WSNs) including security requirements, common attack types, counter-measures against attacks and so on.

2.2.1. Design Goal of Security in WSNs

The primary goal of security protocol in WSNs is to retain the information and resources to be attacked by an adversary. Well designed security services keep the life of Sensor Networks being alive from attacks. So the security scheme designed for WSNs should meet the security requirement discussed below. Generally the design goal of security in WSNs includes [40, 39, 43, 44, 45, 46, 47, 29]:

2.2.1.1. Availability

Security protocol should ensure the availability of networks in terms consume less processing and communication power. Denial of Service(DOS) and node compromise are common attacks in WSNs which results in unavailability of network resources. Therefore, there should be a security protocol, which ensures that the desired network services are available even in the presence of node failure or denial of service attacks.

2.2.1.2. Data Confidentiality

Confidentiality of data is the most relevant issue which makes sure the disclosure of secret information from undesired recipients. It is a critical issue prior to addressed in security requirements of WSNs. In Wireless Sensor Networks, the confidentiality relates to the following issues [41]:

- A sensor network should not leak sensor readings to its neighbors.
- In many application, nodes communicate highly sensitive data, e.g., key distribution, therefore it is extremely important to build a secure channel in a wireless sensor network.
- Public sensor information, such as sensor identities and public keys, should also be encrypted to some extent to protect against traffic analysis attacks.

The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, thus achieving confidentiality.

2.2.1.3. Authentication

Beside to data modification, an adversary can inject malicious packets and can change the whole packet stream. Therefore, the receiver must make sure that data used originates from the right source. Data authentication has enabled the receiver to identify the authenticity of the message and really sent from claiming sender. in WSNs, data authentication realized through symmetric or asymmetric mechanisms where sender and receiver nodes exchange a secret key. For instance, in between two-party communication, data authentication can be achieved via a purely symmetric mechanism: the sender and the receiver share a secret key to compute the message authentication code of all communicated data [41].

A keychain distribution system has been proposed for their μTESLA secure broadcast protocol [45]. The main idea of the μTESLA system is to achieve asymmetric cryptographic by delaying the disclosure of the symmetric key. In this approach first, a sender will broadcast a message generated with a secret key. Then after some period of time, the sender will disclose the secret key. The receiver is responsible for buffering the packet until the secret key has been disclosed. After disclosure, the receiver can authenticate the packet, provided that the packet was received before the key was disclosed.

Generally, authentication service is the insurance for the reliability of the message transmitted in sensor networking.

2.2.1.4. Data Integrity

Maintaining the data confidentiality service is not enough to safeguard the stealing of data by an adversary.Even if the network has confidentiality measures, there is still a possibility that the data integrity has been compromised by alterations. On the other hand, this doesn't mean the data is safe. For instance, the adversary alters the data and can direct sensor networking into disarray. A malicious node may add some fragments or manipulate the data within a packet and can send the new packet to the original receiver. But sometimes the harsh communication environment may cause the damage or loss of data. Obviously, the integrity of the network will be in trouble when [42]:

- A malicious node present in the network injects false data.
- Unstable conditions due to wireless channel cause damage or loss of data.

Therefore, integrity service must ensure that the received data are not being altered or tampered during transmission by an adversary.

2.2.1.5. Authorization

The adversary may deploy malicious nodes within sensor networks to gain the accessibility of network resources [43, 40]. Some time attacker node will perceive its neighbors as legitimate nodes to encrypt or decrypt messages or to obtain full control over networks. Thus the authorization scheme must provide that only authorized sensors can be involved in providing information to network services and authorized nodes can be accessed through network services or resources.

2.2.1.6. Non-repudiation

Most wireless sensor network technology is designed to process sensitive information on the target environment(eg.. military application) . The adversary may control network data by keeping previously sent message to a malicious node or direct to its routing path. Non-repudiation measure must ensure that the reliability of network resources by controlling a node cannot deny sending a message it has previously sent.

2.2.1.7. Data Freshness

The reliability of data is not only assured the integrity and confidentiality schemes, even there is the possibility to the adversary can replay old messages to the receiver. In this case, the adversary may replay expired/old data to network nodes for the purpose of network congestion and be depleting the battery of nodes which results in network performance degradation. Especially considering this requirement during shared-key is very important. Typically, shared keys need to be changed over time. However, it takes time for new shared keys to be propagated to the entire network. In this case, it is easy for the adversary to use a replay attack. Also, it is easy to disrupt the normal work of the sensor, if the sensor is unaware of the new key change time.The freshness problem of data will be solved using include counter value(time-related counter/timestamp) into the packet to ensure the freshness of data. The freshness of data implies that the data are recent, and it ensures that no adversary replay the old messages.

2.2.1.8. Self-Organization

Classically Wireless Sensor Networks (WSNs) are adhoc network, which requires every sensor node to be independent and flexible enough to be self-organizing and self-healing according to different situations. There is no fixed infrastructure available for the purpose of network management in WSNs. This inherent feature brings a great challenge WSNs security. In WSNs,sensor nodes require being independent and flexible enough to be self-organizing and self-healing according to different situations. If self-organization is lacking in a sensor network, the damage resulting from an attack or even the hazardous environment may be devastating.

2.2.1.9. Time Synchronization

Most sensor network applications rely on some form of time synchronization. In order to conserve power, an individual sensor's radio may be turned off for periods of time. Furthermore, sensors may wish to compute the end-to-end delay of a packet as it travels between two pairwise sensors. A more collaborative sensor network may require group synchronization for tracking applications, etc. Therefore, synchronization protocol should realize to secure for pairwise between sender and receiver, multihop sender-receiver and group synchronization within sensor networks.

2.2.1.10. Secure Location

Regularly, the utility of a sensor network will rely on its ability to accurately and automatically locate each sensor in the network [42, 41]. A sensor network designed to locate faults will need accurate location information in order to pinpoint the location of a fault. Unfortunately, an attacker can easily manipulate unsecured location information by reporting false signal strengths, replaying signals.

2.2.2. Classic Attacks in WSNs

WSNs are prone to several types of attack. Due to the high cost of tamper resistant, most sensor nodes in WSNs are deploying and viewed without tamper resistant. So adversary may be able to compromise or physically capture a sensor node to steal key/credential materials included within that sensor node. Also Sensor nodes rely on multihop wireless communication to aggregate and deliver sensed data to the Base Station. Moreover, all vulnerabilities discussed in Sec 2.1.7 make WSNs suffer from several security threats. The aim of the adversary can be the knowledge of confidential information or the knowledge of the significant nodes in the network(cluster head node), by analyzing routing information, to prepare an active attack. Active attacks are a severe attack which can remove or modify the messages transmitted within the network. Besides to these, they can inject malicious traffic or fragment code to disturb the operation of the network or denying the availability of services. Among these, a possible variety of attacks in WSNs are[44, 41, 48, 46, 49, 50, 43]:

2.2.2.1. Denial of Service Attacks(DOS)

DOS attack is any event that diminishes or eliminates a network's capacity to perform its expected function [51]. In addition, the factors like the failure of hardware, software bugs, environmental condition, resource exhaustion or any complicated interaction can cause the denial of services. This Denial of Service(DOS) is performed by malicious actions or unintentional factors(e.g. node failures). The normal DOS attack sends extra unnecessary packets without rule-based protocols to exhaust the resource available at the victim node and thus restricts legitimate network users from accessing network resources or services.

Intentionally the aim of a compromised node is continuously sending messages to overflow the network and to deplete the lifetime legitimate nodes. Several types of DOS attacks might be performed in different layers of Wireless Sensor Networks (WSNs).

The Physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption [52]. Sensor nodes in WSNs use an RF based wireless communication which is broadcast in nature. Because of the broadcast factor, the attacker can generate jam signals to interfere with the radio frequencies used by the sensor networks. Jamming is the main physical layer attack against WSNs. It is a destructive DOS attack which consists of signals that can disturb frequency radio channel by sending useless information on the active frequency band physically. An adversary can disrupt the entire network with $\mathbb{K}$ randomly distributed jamming nodes, putting $\mathbb{N}$ nodes out of service, where $\mathbb{K}$ is much less than $\mathbb{N}$ [51]. This attack is simple and effective for single frequency networks. The jamming situation can be distinguished from neighbor failure nodes by determining that not lack of response, impedes communication and constant energy. There are four classes of jamming attacks [53]:

- Constant Jamming : it involves the complete jamming of the entire network by sending a continuous jamming signal to radio channel disregarding MAC protocols. Thus, no messages are able to be sent or received.
- Deceptive Jamming attack : This type of attack happened when attacker constantly injects regular packets into the channel without any gap between subsequent packets.
- Reactive Jamming : Attacker activates this attack only when there is activity on the channel to severe the reception. Clearly, it stays quiet until there is an activity on the channel then devastates the reception.
- Random Jamming : This alternates between jamming and sleeping mode. Here the attacker(jammer) performs constant or deceptive jammer for a random period of time. This jammer saves energy when compared with others.

All in all, the jammer attacks can compromise sensor networks in order to degrade the network performance by jamming wireless transmissions. Another physical layer attack in WSNs is tampering(Node destruction) [51].It is very challenging to prevents node destruction deployed in an unsecured area. An attacker can tamper with nodes physically and interrogate and compromise them to exacerbates sensor networks. An attacker can extract sensitive material such as cryptographic keys for an undesirable purpose to gain unrestricted access to higher levels of communication. A node can be altered to create compromised nodes, which the adversary uses to control the networks. Defenses against node destruction(tampering) include hiding or camouflging nodes or implementing tamper reaction(such as releasing all programs or cryptographic memory) [51].

MAC protocols operate at link layer which provides arbitrate channel for neighbor-

to-neighbor communication [51]. It ensures reliable point-point and point-multipoint connections for data transfer in communication networks [52]. However, intentionally attacker can violate this communication protocol designed for data transmissions in an attempt generate undesirable situations. A common link layer attacks include [40, 51] : Collision, Exhaustion ,and Unfairness. The adversaries may induce the collision of packets in order to disrupt the entire packets.Once packets collide, a change will likely occur in some portion of data. A change in data portion will cause a checksum mismatch at the receiving end, which results in the packet be discarded as invalid. It can possibly mitigate some collisions using error correcting codes. However, error correcting code may incur the transmission overhead and consume additional energy. Exhaustion : Attacker may trigger collisions repeatedly to cause resource exhaustion.Such collisions would require the retransmission of any packet affected by the collision. Exhaustion attack may exploit the two-way requests Request to Send (RTS) or Clear to Send (CTS) handshake that many MAC protocols(such as IEEE 802.11) use to reserve channel access and transmit data. An attacker can exhaust a node‘s resource(battery) by repeatedly sending RTS messages to elicit CTS responses from a targeted neighbor node. in other words, Using exhaustion attack, it would be possible for an attacker to simply deplete a sensor node‘s power supply by forcing too many retransmissions. One solution used to mitigate this attack is MAC admission control rate limitation, ignore excessive packet requests without sending expensive radio transmissions.A second technique is to use time-division multiplexing where each node is allotted a time slot in which it can transmit. Also, using anti-replay protection and strong link-layer authentication can mitigate this attack. Another link layer threat to WSNs is unfairness. It is the weak form of attack which an attacker causes it in a network by intermittently using the above link layer attacks. This attack may not completely prevent a legitimate node access to service outright or the channel, it degrades a service in order to gain an advantage, such as causing other nodes in a real-time MAC protocol to miss their transmission deadline. One defense mechanism against this attack uses small frames so that an individual node can capture the channel only for a short time.

In multihop sensor networks, the messages traverse many hops before arriving its destination. Since every node is potentially a router to relay a packet, it may lead routing-disruption attacks. Common attacks on routing protocols include spoofing, replaying or altering routing traffic, Selective forwarding, Sybil, wormholes, Hello flood, Black holes and Acknowledgement Spoofing. The adversary may position the malicious nodes within the networks to subvert the network routing protocols and can mount a DOS attack by involving itself in a part of the many packet routes and then lead the dropping all packets (in a *(*black hole attack). Or it may perform Selective forwarding attack i.e..it can selectively forwards packet to reduce the possibility of detection. One way to tackle black holes and selective forwarding attacks is the implicit acknowledgment which makes sure that packets are forwarded as they were sent. However, this technique is reliable only when sensor node‘s radio be active(so consuming power energy). In other words, it is not reliable when

bidirectional links are not guaranteed. Another technique is multipath routing, which sends the same packets over multiple paths to give it a higher possibility of arriving its destination.Nevertheless, multipath routing consumes power on redundant paths and waste additional network bandwidth. Hello flooding is attacker independent attack to compromise security protocols. Many routing protocols which using HELLO messages to inform one-hop neighbors of their presence. An attacker may mount a hello flood by recording hello packets, sending them with high transmit power. In this type of attack, the attacker with high radio transmission range and sends a Hello message to a number of sensor nodes. When assured nodes send messages, then it passes the malicious node as this node provides the shortest path to the base station as an illusion. Then when the messages sent to the malicious node, the victim is betrayed by it which leads data congestion and thus complicates the data flow in the networks. One way to combat hello flooding attack is pairwise authentication, which enables the nodes to verify bidirectional links before constructing routes. A geographic routing protocol such as Geographic and Energy-aware Routing which reduce the hello messages from nodes ,not within communication range and enable sensor nodes to know its location and be able to communicate that location to other nodes. Homing is a network layer attack that investigates pattern traffic to identify geological areas and target nodes that have special responsibilities such as Base Station or Cluster Head. Then an attacker mounts a homing attack to achieve DOS attack by destroying or jamming these key network nodes. A homing attack can be combated by Header encryption, but it does not entirely prevent traffic analysis. Spoofed, Altered or Replayed routing information is a direct attack against routing protocol to target routing information itself while it is being exchanged between sensor nodes. The adversary may mount a DOS attack through spoof, alter or replay routing information in order to disrupt network traffic in the WSNs. This disruption includes the creation of routing loops, attracting or repelling network traffic from select nodes, extending and shortening source routes, generating fake error messages, partitioning the network, and increasing end-to-end latency [40]. Appending a message authentication code is a common countermeasure technique, which enables the receiver to verify whether the message has been spoofed, altered or replayed from malicious nodes. Sybil attack as a malicious device illegitimately taking on multiple identities [54]. It is the attack against network layer presents more than one identity in the network. This attack is effective on routing algorithms, data aggregation, voting and fair resource allocation. This attack can be detected using an identity-based registration approach [54], which make each sensor node check the list of "known good"identity to validate another node as legitimate.

At the transport layer which responsible for managing end-to-end connections, DOS attacks may exploit the protocol that maintains a connection for transmission of information at either end. The two common attacks at transport layer are flooding and desynchronization. In flooding attack, an attacker may create a new connection repeatedly until the resources required by each connection are exhausted which results in legitimate requests be ignored. For instance, in a TCP SYN (*synchronize* flood attack, an adversary sends multiple connection

requests without ever completing the connection, thus overwhelming the target's half-open connection buffer. In a desynchronization, an attacker disrupts active connections between two endpoints by transmitting forged packets, causing missed frames. These messages carry bogus sequence numbers or control flags that cause the endpoints to request retransmission of missed frames [51].In addition to this, an adversary can maintain time correctly, it can prevent endpoints from successfully exchanging any useful information.This attack can be defeated by Header or full packet authentication and authenticate all packets exchanged, plus all control fields in the transport protocol header.

Finally, DOS attack can be performed against application layer protocols. At the application layer, the attacker can mount DOS attack by overwhelming network nodes with sensor stimuli which cause the network to forward a large amount of traffic to the Base Station. Usually, this attacker launch DOS attack at this level to consume network bandwidth and drains power energy. For instance, an adversary can disrupt data aggregation protocol, prevents sensor nodes collecting any aggregated data. Thus DOS attacks let the collecting node to accept the false aggregated value. This type attack can be mitigated by carefully tuning sensor nodes. Rate limiting and secure data aggregation can also mitigate this attack. Generally, table 1 summarizes the possible DOS attacks in each layer of sensor networks with corresponding defense techniques.

TABLE 1. Several types of DOS attacks

Protocol layer	**DOS Attacks**	**Defense Mechanisms**
Physical layer	Jamming	Detect and sleep route around jammed regions
	Tampering	Tamper-proofing the node's physical package
		Hide or camouflge nodes
Data link layer	Collisions	Error correcting codes
	Exhaustion	MAC admission control rate limitation
	Unfairness	Using small frames
Routing layer	Spoofed, Altered, or Replayed Routing Information	message authentication code
	Selective forwarding	implicit acknowledgment
	Sybil	Authentication, probing
	Homing	Header encryption
	Hello flood attacks	pairwise authentication
	Black hole attack	Authorization, monitoring, redundancy
	Acknowledgment Spoofing	Authentication
Transport layer	Network flooding	Client puzzles
	Desynchronization	Header or fullpacket authentication
Application layer	Overwhelming sensors	Sensor tuning and Data aggregation
	Repudiation attacks	Authentication and antireplay protection

2.2.2.2. Node Replica Attacks

Conceptually a clone attack is similar to the Sybil attack, where one physical sensor node gains an unfair advantage by claiming multiple ids. Clone attack is application independent, where one logical node id is reused by multiple physical sensor nodes. Here adversary captures existing sensor nodes in the networks and extracts credential information in the memory to replicating or cloning a new node using the identity of compromised nodes. The aim of this attack is that, cloning both the ID of the original nodes and the cryptographic material used to prove the honesty of the corresponding ID to hide from detection themselves.Using this fashion, an attacker can severely disrupt a network performance: corrupting packets or mislead routing packets. Finally, if the adversary gains physical access to the network it can copy cryptographic keys to the replicated sensor and can also insert the replicated node into strategic points in the network [41]. By deploying the replicated nodes at the specific location point of the network, the adversary can control the entire network by manipulating a segment of a network or disconnecting it all together. Detection of replication attack in WSNs is a particular challenge since every replicated node has a legitimate identity interface. Many replica detection algorithms have been proposed for centralized and Distributed Sensor Networks. In Sec 3 we have presented the detail of clone attack and the overview of the proposed solution for clone attack.

2.2.2.3. Attacks Against Privacy

Privacy problem is relevant concerns because Wireless Sensor technology is a promise for automatic data collection capabilities via efficient deployment of sensor nodes. Concerning these great benefits of Wireless Sensor technology, an attacker can discover the way to abuse the advantages of this technology , specifically the privacy of collecting data [55, 56]. An attacker can use seemingly innocuous data to derive sensitive information if they know how to correlate multiple sensor inputs. For example, in the famous "panda-hunter problem"[55], the hunter can imply the position of pandas by monitoring the traffic. In WSNs, privacy problem is derived from the way sensor nodes collect information from the scattered area. Much information from sensor networks could be probably be collected through direct site surveillance which makes large volumes of information easily available through remote access. Thus the adversaries can easily gather information and monitor multiple sites simultaneously using remote access. Some of the common attacks against sensor node privacy are:

- Eavesdropping : The adversary can simply mount eavesdropping attack by listening network traffic to discover the contents of network communications. If the traffic conveys the control information about the sensor network configuration, which holds potentially more detailed information than accessible through the location server, the eavesdropping can act effectively against the privacy protection[56].

- Traffic Analysis : An increase in the number of transmitting packets between certain nodes could indicate that a specific sensor has registered activity. Through the analysis of the traffic, some sensors with special roles or activities can be effectively identified [56].
- Insert false data : A malicious node could trick the system into reducing data distortion (privacy protection) through spoofing subjects
- Replay Attack : They can insert their node or compromise the nodes to hide in the sensor network. After that, the malicious inserted nodes can masquerade as a normal node to attract the packets, then misroute the packets, e.g. forward the packets to the nodes conducting the privacy analysis

2.2.2.4. Physical Attacks

WSNs usually operate in hostile outdoor environments which is susceptible sensor nodes to physical attacks i.e. physical destruction of nodes. Physical attack is an irreversible attack, it destroys sensor nodes permanently. For instance, an attacker can extract cryptographic secrets, tamper with associated circuitry, modify programming with the sensors or replace them with malicious sensors under the control of attackers. An attacker can substitute the nodes with the illegal and detrimental ones, thus negotiating the functioning of the whole sensor network [50].

2.2.3. Summary

This section described the state of the art in Wireless Sensor Networks (WSNs). To start with the background that describes the applications and architecture of WSNs were analyzed in order to state the characteristics and nature of Wireless Sensor Networks. Then a review of the challenges, design issues, security issues of WSNs were conducted with a revision of the different works literature.

3. Related Work

3.1. Overview

A collection of tiny, low-cost and resource-constrained sensor nodes have enabled of WSNs technology, potentially a solution to real-world challenges. These sensor nodes consist several technologies which make it possible to deploy Wireless Sensor Networks (WSNs). Usually, this type of networks may interact with highly sensitive data or operate in hostile and unattended areas for monitoring and tracking application tasks. Due to this operation nature, WSNs often unattended, hence susceptible different types of attacks. An attacker can capture a node in the existing networks to extract all required information stored in the memory of sensor nodes. Sensor nodes are assumed to not tamper proof. Thus adversary replicates a captured node with large quantity and deploys them under their control in several positions of networks to launch a number of inside malicious activities. This type of attack close to all neighbor nodes to participate in the network operation in the same was legitimate, as they have a legitimate information(ID and cryptographic materials), hence replica node can fire various attacks. The node replica attack might be characterized by two ideas pointed out below:

- Node replica attack is considered as a legitimate node from its neighbor nodes.Indeed, without global mitigation, the legitimate node cannot be aware of the fact that it has a node replica attack among of its neighbors.
- To replicate a large number of compromised nodes, adversary compromise a small of a number of sensor nodes.

Furthermore, several node replica schemes are proposed currently presented in more detail in the next section. Effective schemes for node replica attack detection will save Wireless Sensor Networks (WSNs) from an array of insider attacks launched in the network. The detection of node replica attacks is either network based or radio-based detection. Radio-based detection uses the strength of radio signal utilized at the receiver node to detect node replication attack [57]. However, the scheme is unfortunately impractical for unattended and geographically widespread WSNs. Herein, we focus on network-based detection of works done before on node replica.

A Wireless Sensor Networks (WSNs) can be either mobile or stationary, clearly summarized in table 2. In stationary WSNs, the sensornodes are deployed with a fixed position in the network deployment. In other words, after deployment sensor nodes do not change their positions. In other hands, in Mobile Wireless Sensor Networks (WSNs), all sensor nodes are roaming in the sensor fields all the time and they are interacting with environmental phenomena by controlling their movement, thus stationary strategy is not applicable herein. Clearly, this indicates that mobile nodes are able to reposition and organize themselves

TABLE 2. Mobile and Stationary WSNs features

	Stationary WSNs	**Mobile WSNs**
Network nature	Sensor nodes are static	Sensor Nodes are roaming in the network at any time
Routing	Use fixed routing to data distribution and aggregation	Use dynamic routing
Deployment	Random and Unchanged deployment position of Sensor Nodes	Mobile and Sensor Nodes able to reposition their deployment location at any time
Communication	Communicate to each each other only when they in the same a range of communication and exchange gathered information. In this case, one node can communicate with many other nodes at different times.	In this approach, sensor nodes keep communication remain within other nodes in the same range until application services achieved
Replication Detection	Based on Node Location	Involves different scenarios and techniques
Topology	Unchanged Network partitions	Frequently changed Network Partitions

in the network and spread out to collect information. Node Replica detection in mobile WSNs involves different scenarios and techniques. Usually, in mobile WSNs, node replica attack detected using witness nodes [58, 59, 60]. In this approach, each node broadcasts its location and identity information to its neighbor. Then it will send the node‘s information a randomly selected node called witness node. Witness node detects a node replica attack by checking the identity (ID) with its location. However, the replicated node may continue to follow this protocol. Meanwhile, the adversary can drop or suppress messages of identity (ID) and location claims to avoid detection of replicated nodes or clone attacks to extend its lifespan within the networks. Beside of that, due to message overhead, these schemes cannot detect node replica attacks in an actual time of attacking, thus they could not completely protect the network from node replica attacks. Furthermore, these schemes are applicable only when the number of node replica attacks are limited.

Table 2 depicts that stationary and mobile WSNs differ in their features which signifies replication schemes for stationary and mobile WSNs will be significantly different. Approaches detecting for node replication attack in stationary and mobile WSNs are further divided into two classes, namely, Centralized and Distributed schemes. This classification offers the better understanding of node replication detection schemes. Hereinafter, we only consider focus on node replication schemes proposed for stationary WSNs in centralized approach because of the proposed scheme designed in a centralized approach.

3.2. Existing Node Replica Detection Schemes

To alleviate the effects of node replication attacks, a number of security schemes have been proposed in either centralized or distributed fashion in stationary WSNs. The comprehensive and detail existing replication detection schemes for stationary WSNs have been discussed herein Sec 3.2.1 which has pointed out some of technical advantages and weakness.

3.2.1. Centralized approaches

In central schemes, Base Station supposed to be a powerful node responsible for information convergence and monitoring every task and communication carried out between Sensor nodes. Moreover, BS has the capability for decision making, thus Base Station (BS) accountable to identify the identity of nodes and detect node replication attack. In these existent schemes, the two key solutions used to detect node replication attack are a list of deployed sensor nodes and their claim locations. In other words, throughout replication process, each sensor node sends in the networks sends its location claim ($ID, Location\ Info$) and a list of its neighbors with their claimed location to the Base Station (BS) [59]. Based on the receiving location claims, the Base Station (BS) perform crosscheck the node IDs along with their corresponding locations, and if the same logical ID found from different locations will result in node replication detection. Finally, Base Station (BS) triggers node replication alarm.

In this section, we discussed the intrinsic centralized replication schemes with their contributions and respective limitations. Here, we have detailed the existent node replica detection schemes:

3.2.1.1. Coned keys Detection

This scheme detects a node replication attack using key usage based strategy by sensor nodes. In [61], a node replication scheme has been proposed in the perspective of Random Key Predistribution. The key assumptions and application setting in this approach are something different from others; actually, instead of cloned sensor nodes, it concentrates on the detection of cloned cryptographic keys which categorized into anomaly detection. The basic idea behind in the context of random key predistribution is that every legitimate sensor node should follow a certain pattern. Hence, it is feasible to control and monitor key usage, which realizes how many times one key used to establish a secure connection between neighboring sensor nodes rather how many a key used to encrypting or decrypting a packet. The approach detects clone attacks by analyzing node authentication statistics to figure out those keys whose usage exceeds a threshold which indicates statistical deviations to detect cloned attacks and then remove from the network. The protocol uses counting Bloom filters to collect key usage statistics, performed by the Base Station (BS). This

protocol allows each node to append a random number to Bloom filter and encrypts obtained results with the public key of the BS and forwarded its own filter results to the Base station, which then performs statistical deviations to discover node replication attacks. The Base Station decrypts the Bloom filter it receives and counts how many each key is used in the network. If the usage of the key exceeds the threshold value, then it is detected as a replicated node. A Base Station (BS) performs a bloom filter from the replicated keys, encrypts the list of keys using its secret key and then broadcast this filter to the sensor networks using a gossip protocol. To this end, each sensor node decrypts bloom filter of Base Station (BS) to remove replicated keys from its keying disconnects connection using replicated nodes.

However, it seems to be that this detection of scheme effective when :

- the size of the pre-loaded keys to each node is small
- more clones exist in the network
- a high false positive rate is set

This condition indicates that possibly poor detection accuracy, i.e. high false negative and positive rates for real scenarios. Beside of this, it is assumed in the scheme that connections between all nodes are likely equal, however, practically in Wireless Sensor Networks (WSNs) , any sensor node can only communicate with a limited number of neighboring nodes within a finite wireless communication range. How to ensure that the participating clones report their keys honestly and exactly to the BS is another limiting of this scheme overlooked.

3.2.1.2. Social Fingerprint verification based Scheme

In [62], real-time detection of clone attack has been proposed using encoded social community information called social fingerprint. This approach falls under the category of neighborhood social signature-based techniques. The scheme incorporates two phases to detect clone attack:

1) In the 1^{th} phase, each sensor node computes a fingerprint by incorporating the neighborhood information which using superimposed s-disjunct code and store the fingerprint of all neighboring nodes which reflects the characteristics of neighborhoods.
2) In the 2^{th} phase, each time a sensor node sends a message, the node should send a fingerprint along with a message to verify the legitimacy of message originator and detection is conducted at sensor side and at BS. If there is replica node deployed in another position of the network, sends a message consists of fingerprint belongs other community will be detected and dropped because a replica node attack can have same credential info (ID, keys) , but it does not belong to the same "community ".

The scheme explores two key assumptions to achieve clone attack detection : sensor deployment topology and social characteristics between each sensor node. It realizes that, once sensor nodes are deployed, they are positioned with a fixed neighborhood(unchanged topology). The sensor nodes form a small "community "or "social network "with its neighborhood. A cloned sensor can have the same legitimate credentials (ID, keys, etc.) as the original node, but cannot have the same community neighborhood. Thus, each sensor can be distinguishably characterized by its social community network. In a small community, a newcomer can be easily recognized if speaking with a different accent. Likewise, a replica node can be easily identified by its neighbors if carrying a "social signature "belonging to a different community.

Nevertheless, this scheme was based *absolutely* fixed WSNs, thus neither node addition nor disappearance can be handled. In addition, it cannot handle a sophisticated replica which can cleverly compute by itself a fingerprint consistent with its neighborhood so as to flee the detection at the sensor side. A more intelligent replica can escape and avoid the detection at the base station simply by not communicating with Base station. It also computationally expensive to generating code-word from superimposed s-disjunct code for each sensor node.

3.2.1.3. SET operations based techniques

Another centralized based node replication detection has been proposed known as a SET. In SET, the network is randomly divided into exclusive subsets. The SET is an operation based scheme which attempts to compute SET operations(Intersection and Union) of exclusive subsets in the network to reduce detection overhead. SET logically partitions the network into exclusive subsets (clusters) respectively managed by leaders(cluster heads) and has the leaders responsible for reporting to the BS all the IDs of nodes in the cluster in the form of the subset. The intersection operation is performed on the root of each subset by BS. It detects a node replication attack if the intersection of two or more subsets is nonempty, otherwise, there is no clone node detected in the subsets.

Although SET performs multiple rounds to counter colliding replicas which cause higher detection cost in terms of computation and communication. Moreover, unexpected design flaws can make adversary to misuse the detection protocol to revoke legitimate nodes [63]. Indeed, it can be exploited by adversary malicious activities to revoke nodes that are not malicious (honest nodes). In addition to that, the malicious node can act as subset leader and could declare honest node, say α, exists in some part of sub-tree or network (that is belonging to another sub-tree) and leads the network to detect and revoke honest node α. Due to the possibility of this attack, a SET protocol is not satisfactory to detect node replica effectively.

3.2.1.4. Cluster Head Based Scheme

Cluster-head selection-based hierarchical distributed scheme is outlined in [64] for detecting node replication attacks using a Bloom filter mechanism including the network reactions. Furthermore, the scheme relies on a cluster head selection performed using the Local Negotiated Clustering Algorithm (LNCA) protocol [65]. Finally, each Cluster Head(CH) exchanges the member node IDs through a Bloom filter with other Cluster Head (CH)s to detect node replications. This scheme has three phases:

① *Predistribution phase* : At this phase, the BS generates the required cryptographic materials and store them in the memory of each sensor nodes.

② *Election phase* : The Cluster Head (CH) election is carried out using LNCA protocol.

③ *Detection phase*: In this scheme, replicated nodes detection is performed using a CH.

For instance, a CH computes a dynamic Bloom filter to build the list the node IDs of its cluster members including itself. Then sends the messages to other Cluster Head, CH, with its authentication keys and CH authenticate one or more node IDs from the list of receiving IDs to build a new Bloom filter. Finally, Bloom filter construction is performed by each cluster head, and the Bloom filter verification is performed by the other cluster heads. If a new created Bloom filter is the same as the first one, the Bloom filter is accepted and the verification begins. However, this type scheme computes high communication and storage cost because this scheme detects clone attack by exchanging cluster member's information between CHs.

3.2.1.5. Zone Based Scheme

Mishra et al [66] have proposed a scheme based on the dividing the network into several zones for the detection of node replica attack. In this scheme, each zone has the zone leaders, which designated to detect node replica attack within the networks. Zone leaders broadcast the registration to each node after deployment, then, they perform the registration and send back the zone join message to the zone leader which is closest to it. This protocol detects a node replica attack in two levels. In the first level (intra-zone detection), it performs detection if a newcomer or new nodes want to join the zone after all nodes finish registration. The zone leader will receive the join message of a new node and check its identity in the member list of its own zone and if the ID is already in its own member list, the zone-leader will broadcast a zone revoke message for the replica node and remove the replica node from the network immediately. Otherwise, if the ID does not exist in its own member list zone, it will go to the second level (inter-zone detection) check i.e. the zone leader requested for joining is send the join message to another zone leader to check whether a node ID exists in another zone member list or not. If the two-level detection confirms no ID matching, the node will join the network and added to the

member list. This scheme has significantly minimized the storage overhead since each node does not need to store any location claims in their memory. However, this scheme is based on the trusty zone leader. Meanwhile, the network will be compromised when the adversary replicates the zone leader. In addition that, this scheme does not detect a node replica attack in the real-time because of finding of attack ID in the member lists of all zone leaders which will bring the communication overhead in another way.

3.2.1.6. Compressed sensing-based scheme

In [67], BS based technique node replication protocol has been proposed called compressed sensing-based clone identification (CSI) for stationary wireless sensor networks. CSI let every node broadcasts a fixed sensed data (α) to its one step neighbors. Then sensor nodes forward and aggregate the received numbers from descendant nodes along the aggregation tree through sensing-based data gathering techniques. In this scheme, the Base Station (BS) act as the aggregation tree to receive the aggregated result and recovers the sensed data of the network. Here one threshold value is fixed (α). If the sensor node has the sensor reading greater than the fixed value, then that node is judged as a cloned node. They have used a novel concept of compressed sensing for the identification of clones in the sensor network. This scheme has the lowest communication overhead comparison with other schemes. Base Station (BS) is responsible for the aggregation of the result (decision) about the identification of clones in the network. But this scheme detects clone attack if data is forwarded has been forwarded to the BS from node replica nodes.

3.2.2. Summary

In this section, we have discussed the existent node replica detection schemes in order to highlight their contributions towards the concept of node replica detection. The performance of these schemes was presented and evaluated leading to a critical analysis. Based, on the literature review a clear pattern of node replica detection of the existent schemes are based on the concept $\mathcal{ID}$ and location claims of sensornodes. However, the program code in a clone sensornode has been programmed by the adversary for the purpose of escaping from being detected. In the other word, the adversary can modify the clone sensornodes for the purpose of colliding the clone sensornode with the compromised sensornode, which will lead to unsatisfactory of existing schemes to detect cloned attacks in WSNs.

In this work, we have proposed detection scheme against clone attack with improved communication cost, network lifetime, detection probability and energy consumption than previous works. We proposed a node replica detection scheme to overcome the drawbacks of existing schemes. In our scheme, node replica detection is not only based on the sensornode's $\mathcal{ID}$ and location claims, we have proposed a new mechanism which has been discussed in Sec 4.

The rest of the paper is organized as follows: The proposed scheme is described and analyzed in Sec 4. Simulation results and Analysis are given in Sec 6, and finally few conclusions and future work direction are drawn in Sect 7.

4. Proposed Security Models and Frameworks

5.1. Overview

As discussed in Related works, Sec: 3, the extant schemes are not satisfactory to detect node replica attacks in Wireless Sensor Networks (WSNs) due to factors like network density, communication overhead, nature of network operations, etc. in both distributed and centralized WSNs. For instance, two detection schemes proposed in [58, 59, 60] , use the witness node to detect node replica attacks upon collecting location claims of sensor nodes, requires a high amount of information transmission, which results in message overhead and consequently low capacity to detect node replica attacks. In addition, these schemes are not suitable in large areas of WSNs to detect node replica attacks in the exact time occurrence of an attack.

Specifically, the existing detection schemes solutions in Cluster-Based Wireless Sensor Networks (CBWSN) have their own limitations to secure an adversary subvert to the Cluster formation intended for node clustering and CH election. And also securing CH role Rotation performed during normal operation of Networks. Therefore, in this thesis, we proposed the detection scheme in Cluster-Based WSNs which detects and expel node replica during the operations of cluster formation. Specifically, the proposed scheme secure the two most important operations of cluster formation: CH election and CH role rotation of Cluster-Based Wireless Sensor Networks (CBWSN) .

The rest of this section is organized as follows: In section 5.2, the architectures of the proposed scheme are briefly presented. Finally, the proposed scheme detection on node replica and security analysis are introduced in Section 5.3.

5.2. Architectures of Proposed Scheme

This section is introduced the overall design assumptions and architectures used in the proposed scheme for detection of clone attacks during node clustering in CBWSN. In Section 5.2.1, the overall the assumptions of the proposed scheme are discussed.

5.2.1. Assumptions

In this section, we present out the assumption used in the proposed scheme. In section 5.2.1.1 and section 5.2.1.2 the network and adversary model are discussed.

5.2.1.1. Network Model and Assumption

In this paper, we assumed that hierarchical static homogeneous Wireless Sensor Networks, which divides network operation into three-tier architecture: BS, CH ,and constrained

sensor nodes. First, we assume that sensor nodes $\mathcal{V}$ scattered in the observation area. After deployment, sensor nodes are form secured clusters. Once $\mathcal{N}$ nodes partitioned into the clusters, each cluster member establishes a pairwise with only other cluster members in the same cluster. Thereafter, all nodes $\mathcal{V}$ inserts themselves in CH candidates. Our schemes revoke node replica attacks from Cluster Head candidates. Next, our detection scheme selects one node as CH for each cluster among trustworthy cluster members based on CH election threshold value and the selected CH announce its head role to cluster members. Moreover,we consider that there exists $\mathcal{N}$ sensor nodes $\mathcal{V} \in \{v_1, v_2, v_3, \ldots, v_{\mathcal{N}}\}$ and Clusters $\mathcal{K} \in \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \ldots, \mathcal{K}_{\mathcal{N}}\}$ in the network. Let $\mathcal{V}_i$ and $\mathcal{V}_j$ are neighbor nodes exists in $\mathcal{K}_i$ and $\mathcal{K}_j$ clusters respectively, thus we can deduce that $\mathcal{K}(\mathcal{V}_i)$ and $\mathcal{K}\left(\mathcal{V}_j\right)$ are different since $\mathcal{V}_i \notin \mathcal{K}\left(\mathcal{V}_j\right)$ and $\mathcal{V}_j \notin \mathcal{K}(\mathcal{V}_i)$, but $\mathcal{V}_i \in \mathcal{K}(\mathcal{V}_i)$ and $\mathcal{V}_j \in \mathcal{K}\left(\mathcal{V}_j\right)$.

Then, the sensor nodes sense the target data from observed area and forwards it to the CH and finally, the collected and disseminated data are transmitted to the BS in as depicted in Figure 20. These two-time slots (CH election and data transmission) are recycled until the end of the CH role round and rotated by another node securely. The sensor nodes can maintain intracluster communication with their CH directly.

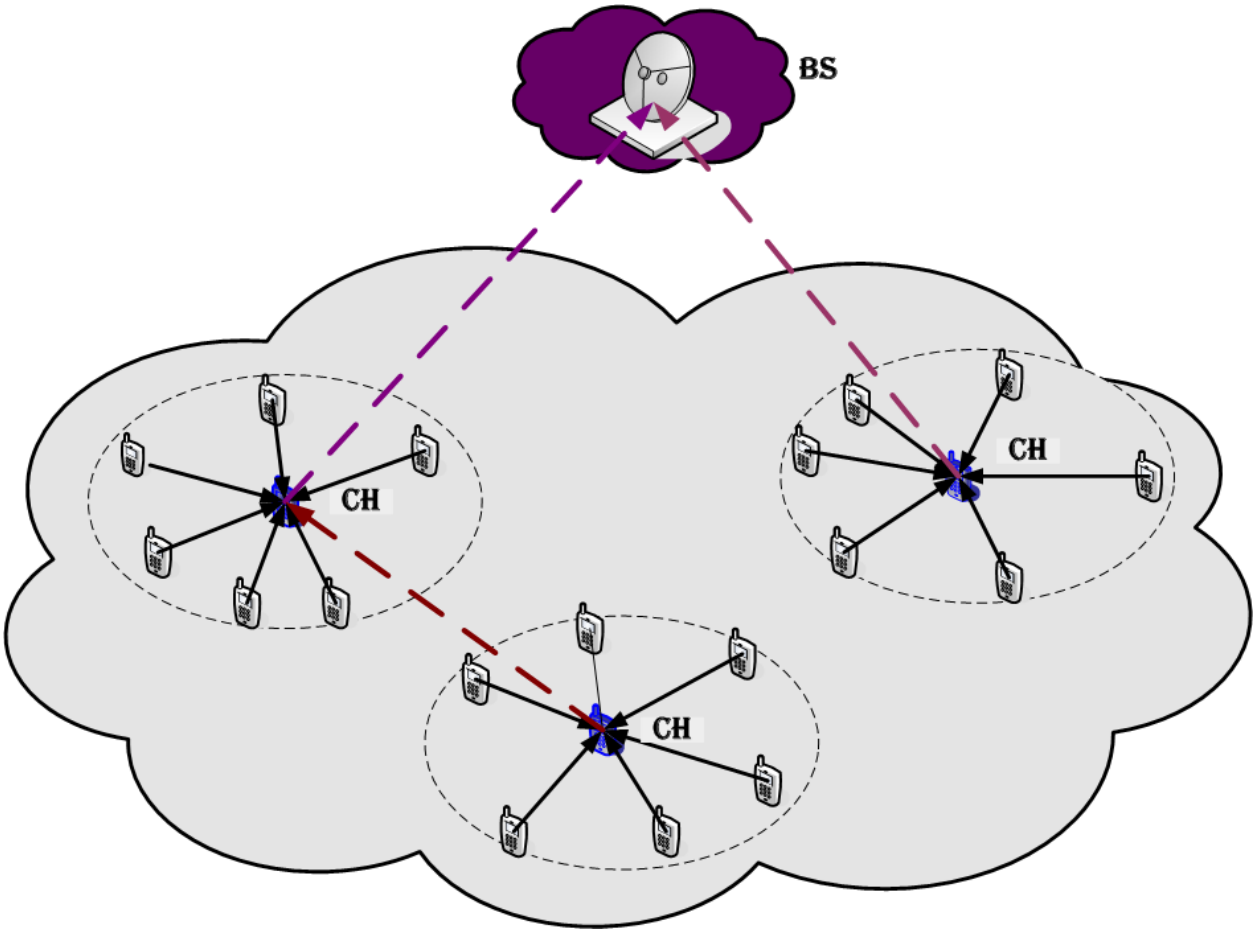

Figure 20. Assumed Network Model

Moreover, we consider the following assumptions for the network architecture of the

proposed scheme:

- Base Station (BS) is powerful and located in a fixed position far away from the sensor nodes.
- Sensor nodes are assigned with a unique credential prior to deployment.
- The positions of sensor nodes are determined after deployment using self-localization schemes proposed in [68].
- All nodes can participate in cluster formation in being elected as Cluster Head (CH) based on predefined threshold values.
- One sensor node belongs to only one of a cluster.
- The CH role rotation has occurred when the energy level of current CH below the predefined energy threshold.
- The network topology will be reclustered, if more than threshold nodes are dead within a cluster.

5.2.1.2. Attack Model

In this paper, we consider the Base Station (BS) and Cluster Head (CH) are powerful nodes to detect node replica attacks deployed by adversary within the networks. In our case, Sensor nodes are not tamper-resistant and deployed in unsecured areas. So, an adversary can easily capture some legitimate sensor nodes from sensing area to release all its credential information. Afterward scanning all security information(e.g. ID,keys,etc) of compromised nodes, replicates them and deploy in different positions of networks. Then clone nodes can easily take part in node clustering operations to being elected as CH and rotate the CH role in the same fashion as legitimate nodes.

In fact, the node replica attacks are monitored by an adversary, so that the cloned node is assumed to broadcast a CH role to the Cluster Member (CM) since the node replica attacks have all credential information of compromised nodes. The node replica attacks announce cluster-head status with a high strongest signal to all nearest neighbor nodes. So, it can obtain joining request messages from legitimate nodes and can keep maintaining the Cluster Head (CH) role until it is detected as node replica attack and expel out of the networks. Moreover, using these clone nodes the adversary can launch several internal attacks such as false route information, replay expired/false event report, reporting false signal strength, maintaining inject bogus data to depleting sensor node's battery, etc.

Moreover, the adversary can obtain a large amount of data packets from neighbor nodes illegally. Because the clone nodes behave as a legitimate node to neighbor nodes. For that reason, we have proposed a new scheme that protects CH election and role rotation from clone attacks that have been discussed in Section 5.3 in detail.

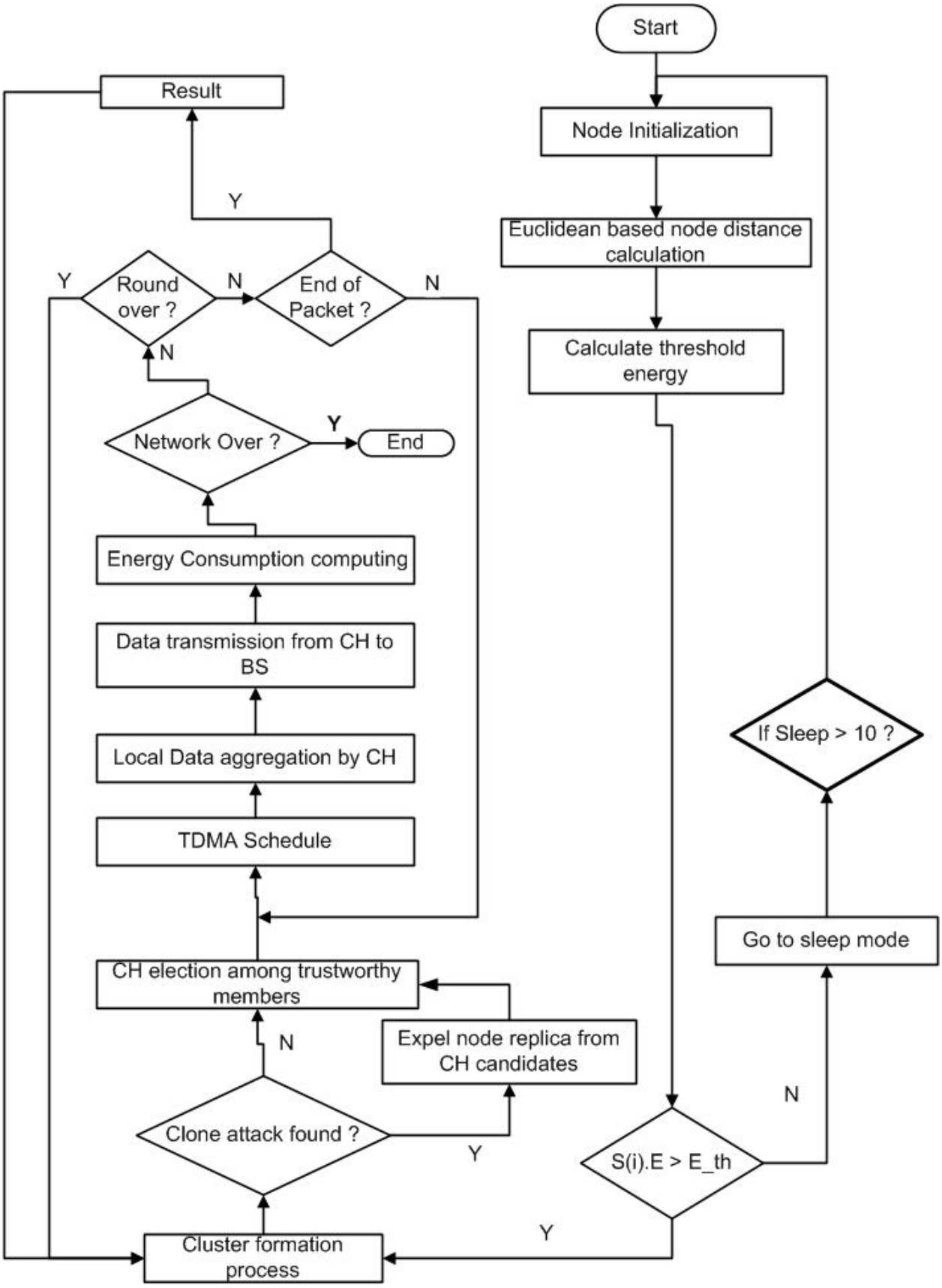

Figure 21. Flowchart of Proposed Scheme against Node Replica attacks

5.3. Torching Node Replica detection in CBWSN

In this section, we introduced and described the proposed scheme against node replica attacks in detail. For the sake of simplicity, we summarized the notations and symbols used throughout the thesis in Table 3.

TABLE 3. Symbol Definition

Notation	Definition
$\mathcal{V}$	$\{v_1, v_3, v_3, \ldots, v_N\}$ (a set of nodes to be clustered)
$\mathcal{N}$	Total number of Sensor nodes
$\imath$	Number of Clusters and Nodes
η	The assumed number of sensor nodes in each cluster
$\mathcal{P}_{stn}$	Geographical of position Sensor node $\mathcal{V}$
$\mathcal{E}_r$	Energy level of Sensor node $\mathcal{V}$
$\mathcal{T}(e)$	The probability energy threshold value of sensor node $\mathcal{V}$
$\mathcal{R}_n$	Number of round sensor node $\mathcal{V}$ stay with a CH role
$\mathbf{C}_{rng}$	Communication range
r	Minimum communication range
$\mathbf{A}$	Network area

The proposed scheme divides the network into clusters and makes one node being elected as a CH in each cluster. However, in the proposed scheme, one node cannot be either Cluster Member or Cluster Head in two clusters located with different positions. Our goal is to enable any of the legitimate nodes $\mathcal{V}_i$ to participate in CH candidates in order to become eligible CH without the order of an adversary. CH is selected between sensor nodes after clustering nodes. The security life cycle of the proposed scheme consists five phases as depicted in figure 22. These five cycles have classified the operations of the proposed detection scheme in to two: Cluster formation and Node replica attack detection.

The next sections discuss how the proposed scheme detects node replica attacks during cluster formation and prevent it from being elected as CH in cluster-based WSNs architecture.

5.3.1. Cluster Formation

A cluster formation of the proposed scheme consists of three phases: **Pre-Deployment, Pre-Clustering, Node-Clustering**

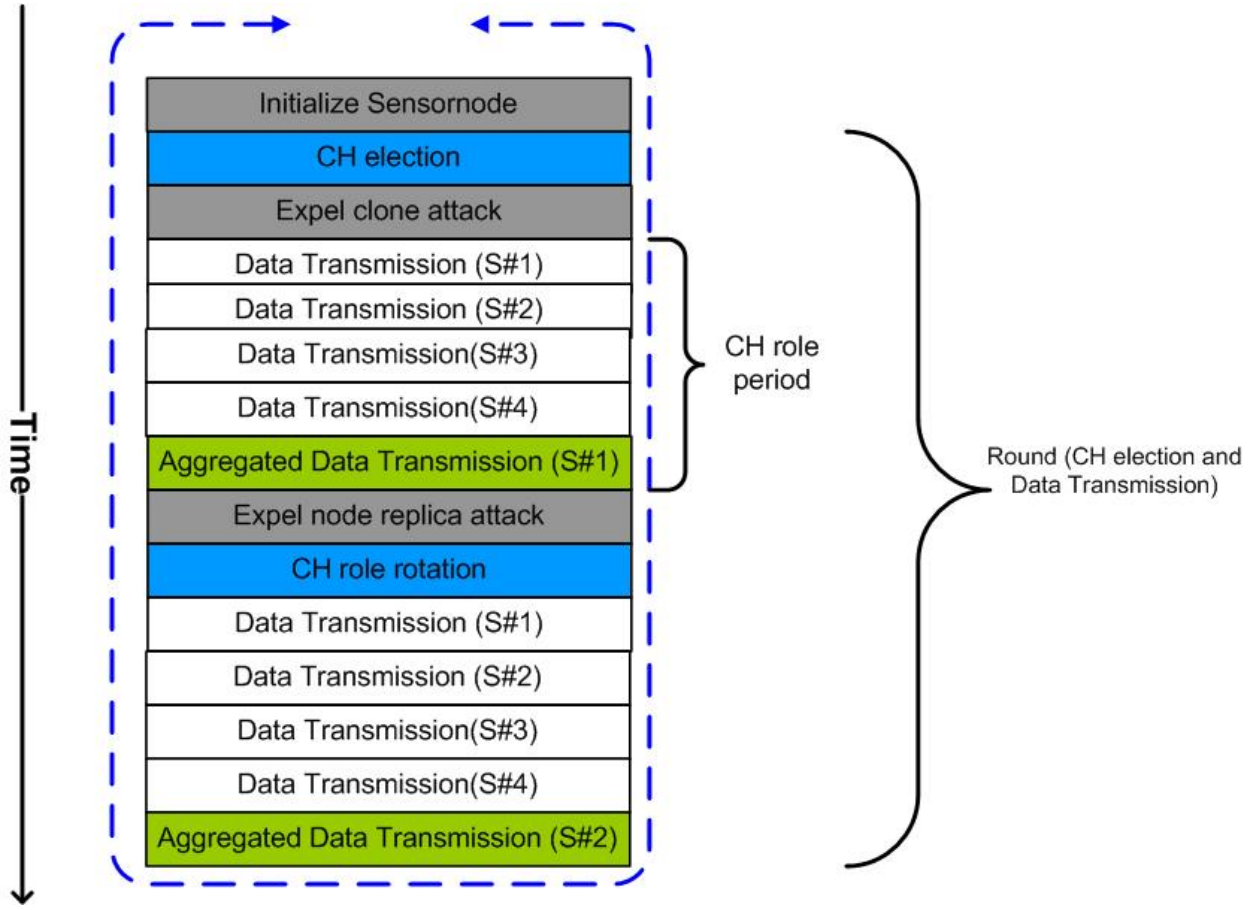

Figure 22. Operation life cycle of the Proposed scheme

5.3.1.1. Node Initialization

Before the deployment of the sensornodes in the $\mathcal{A}$ are, the BS generates network parameters and registers the sensor nodes by including it in a $\mathcal{N}$ $Sensor\ node\_lists$

The BS takes a security parameter $\delta \in \mathbb{Z}^+$ as inputs and generates ϕ network security parameters as follows: Given the δ Security parameter, generate a prime number q, two Abelian cyclic groups $\mathcal{G}_1$ and $\mathcal{G}_2$ of order q such that $\varphi : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ and arbitrary generator $g \in \mathcal{G}_1$.

Finally, each $\mathcal{V}$ sensor node within the networks is preloaded with ϕ_{params} public security parameters before deployment. The BS responsible for generating ϕ public security parameters to each $\mathcal{V}$ sensor node such that the BS computes algorithm 1.

Algorithm 1 Algorithm of Network security parameters Generation

INPUT: δ Security parameter

function NODE_INITIALIZATION(δ)
  Select Security parameter $\delta \in \mathbb{Z}^+$
  Generate a prime number q
  Compute two cyclic groups $\{\mathcal{G}_1, \mathcal{G}_2\}$ with prime order q
  Compute Bilinear pairing φ: $\mathcal{G}_1 \times \mathcal{G}_2$ such that $g \in \mathcal{G}_1$ be a generator
  Compute the hash function $\mathcal{H}_1 : \{0,1\}^\tau \mapsto \mathcal{G}_1^*$ ▷ compute of $\mathcal{H}_1$ hash function of τ given by $\mathbb{H}_1\{0,1\} = \mathcal{G}_1$
  Compute the function $\mathcal{H}_2 : \mathcal{G}_2^* \mapsto \{0,1\}^n$ ▷ Compute $\mathcal{H}_2$ hash function of $\mathcal{G}_2$
  Compute Bilinear $\mathcal{H}_3 : \{0,1\}^{n+\tau} \times \mathcal{G}_1 \rightarrow \mathbb{Z}_q^*$
  Compute Network Private codeword $N_{sk} := \overset{\mathcal{R}}{\leftarrow} \mathbb{Z}_q$ ▷ Randomly Select N_{sk} from the polynomial time results
  Assign N_{sk} to the Base Station: BS $\leftarrow N_{sk}$ ▷ N_{sk} is kept secret
  $N_{pk} := g^{N_{sk}} \in \mathcal{G}_1$ ▷ Generate Public Network Key N_{pk} for $\mathcal{V}$ node
  Security parameter $\phi_{params} = \{\varphi, \mathcal{G}_1, \mathcal{G}_2, g, N_{p_k}, \mathbb{H}_1, \mathbb{H}_2, \mathbb{H}_3, q, n, \tau\}$
  Determine a number sensor nodes , $\mathcal{V}_i = \{v_i | v_i = \mathcal{ID}_i, i \in [1, \mathcal{N}]\}$
  while $i > 0$ **do**
    BS $\rightarrow \mathcal{V}_i : \phi_{params}$ ▷ BS assign ϕ_{params} to each sensor node
    $i + 1$
  end while
  return $\mathcal{V}_i$ preloaded with ϕ_{params}
end function

Hereafter, BS store $\phi_{params} \times \mathcal{N}$ assigned to each $\mathcal{V}$ sensor node and preload each sensor node with ϕ_i corresponding to $\mathcal{ID}_i$ while N_{sk} is kept secretly with BS. Afterward, all sensor nodes deployed with unique codeword in the observation area $\mathcal{A}$. Finally, every sensor node locates its physical neighbors within its communication range.

5.3.1.2. Pre-Clustering Phase

In this scheme, the communication carries out between sensor nodes during cluster formation is based on computing message $\mathcal{M}$ and trapdoor signal $\mathcal{S}_\tau$ mechanism. The BS responsible to determine the right time communication among sensor nodes using trapdoor signal mechanism. Prior to clustering, the BS publishes trapdoor signal which enables sensor nodes to identify real-time communication and consequently used to detect node replica attack during lifespan operations of WSNs, particularly during cluster formation. To publish trapdoor time signal, the BS takes N_{sk} computed in algorithm 1 and τ current time information as inputs and then generates trapdoor time signal $\mathcal{S}_T$ as output.

Algorithm 2 Algorithm of trapdoor Time Signal $\mathcal{S}_T$ generation.

INPUT: N_{sk} and $\mathcal{T}$

function TPDoor(τ, N_{sk})
  $\mathcal{T} = \mathcal{H}_1(t)$ ▷ Compute hash function of current time $\mathcal{T}$, where $\mathcal{T} \in \{0,1\}^\tau$
  $\mathcal{S}_\tau = \mathcal{N}_{sk}\mathcal{T}$
  return $\mathcal{S}_\tau$
end function

Right after the deployment, the BS divides the lifetime of WSNs into Cluster-formation phase and data aggregation phase. During cluster formation phase, the proposed scheme detail into two things: BS partitions the networks into clusters based on the assumed threshold and detect node replica attacks. It is supposed that every node deployed in sensing must participate during the cluster setup process to prepare it for CH candidates. At the end of cluster formation setup, Cluster Head (CH) and cluster members are two classes of nodes delivered as to determine the lifespan and accomplish the sensing role of Wireless Sensor Networks (WSNs). However, in this scheme, we consider that each sensor node must obtain special consistent codeword of this scheme known as **cluster_formation_Social_(CF)** codeword. The **CF** declares the permission for each sensor node to be a member of the cluster formation setup process for being elected as either CH or be Cluster member. Prior to that, each sensor node should be preloaded with ϕ_{params} security parameters conducted by BS in algorithm 1. The construction of **CF** computed among sensor nodes after deployment and prior to the cluster formation. All sensor nodes use ϕ_{params} as input and construct **CF** by computing algorithm 3 as follows:

Algorithm 3 Algorithm of Cluster Formation codeword Generation for Sensor nodes

INPUT: ϕ_{params}

function $\mathbf{CF}_{codeword}(\mathcal{V}, \mathbf{CF}_i^v)$
  Determine number of legitimate nodes $\mathcal{V}_i = \{\mathcal{ID}_i | i \in [1, \mathcal{N}]\}$
  for $\forall\ \mathcal{V}_i \in \mathcal{N}(\mathcal{V})$ **do**
    Compute the private **CF** key of $\mathcal{V}$ node : $\mathbf{CF}_{i_{sk}} \mapsto \mathbf{Z}_q^*$
    Output the public **CF** of node $\mathcal{V}_i$: $\mathbf{CF}_{i_{pk}} \leftarrow g^{(\mathcal{ID}_i)\mathbf{CF}_{i_{sk}}}$
  end for
  return $\mathbf{CF}_{i_{pk}}$, $\mathbf{CF}_{i_{sk}}$
end function

Thereby, all sensor nodes in the clusters share **CF** key with cluster members. The **CF** key is mainly used for securing status message in a cluster and to identify cluster social community. Only the CH can update the **CF** when $\mathcal{V}$ sensor node dead or join the cluster.

5.3.1.3. CH election

In the proposed scheme, the BS considered as powerful and has unlimited energy. Thus, the BS has no power consumption constraints to processing network data and controlling

the communication between sensornodes. For the purpose of optimal CH election, a number of sensor nodes, geo-location ,and energy level of nodes used as input parameters in this scheme. The scheme achieves optimal CH election and role rotation by measuring the current energy level of nodes and the positions of $\mathcal{V}$ nodes from the BS. The energy level of each sensor node is calculated by using the energy threshold and also the position reference of nodes $\mathcal{V}$ from the BS is measured by Euclidean distance [68]. The BS aware the locations of all sensor nodes through GPS mechanism or another mechanism [69].

Afterward, each sensor node broadcasts a status message $\mathcal{M}_{\mathcal{EP}} \in \mathcal{M}$ consisting of energy level $\mathcal{E}_r$ and location $\mathcal{P}_{stn}$ to the BS directly. The broadcasting of energy level $\mathcal{E}_r$ and positions $\mathcal{P}_{stn}$ of sensor nodes are conducted until the nodes in the networks broadcasting information about their status and delivered to the BS. The transmitted message should be in the following format:

$$\mathcal{V}_i \rightarrow BS : \left[\mathcal{M}_{\mathcal{EP}}\{\mathbf{CF}_{pk}, \mathcal{N}_{pk}, \mathcal{P}_{stn}, \mathcal{E}_r\}\right]$$

First, after receiving the status message, BS divides the networks into a small area called clusters based on geographical locations, the distance of sensor nodes from BS and minimum communication range between sensor nodes. The estimated number of sensor nodes in one cluster can be considered as:

$$\eta = \frac{N \times \Pi \times r^2}{\mathcal{A}} \tag{2}$$

Thus, the total number of sensor node $\mathcal{V}$ in one Cluster $\mathcal{K}$ can be:

$$\mathcal{K}(v) = \sum_{i=1}^{\eta} \mathcal{V}_i \tag{3}$$

Next, BS selects optimal CH for each cluster based on $\mathcal{M}_{\mathcal{EP}}$ transmitted from each sensor node. The BS calculates the Energy level $\mathcal{E}_r$ of $\mathcal{V}$ sensor node using an energy model adopted in [33, 5] and in order to select more energy efficient CH election per each cluster. The initial energy level $\mathcal{E}_r$ for each sensor node is same. Thus, the energy dissipated by $\mathcal{V}$ sensor node to transmit or receive $\mathcal{M}$ to/from its Cluster Member or neighbor nodes $\mathcal{N}(\mathcal{N}(\mathcal{V}))$ is adopted on equation (4) and equation (5) over d distance:

- Energy dissipated by $\mathcal{V}$ sensor node for transmission of $\mathcal{M}$ over d distance:

$$\begin{aligned} \mathcal{E}_r(\mathcal{M}, d)_{\mathcal{T}_{trsm}} &= \mathcal{E}_{\mathcal{T}_{trsm\_elec}}(\mathcal{M}_i^v) + \mathcal{E}_{\mathcal{T}_{trsmt-amp}}(\mathcal{M}_i^v, d) \\ &= \mathcal{E}_{elec} \times \mathcal{M}_i^v + \mathcal{E}_{amp} \times \mathcal{M}_i^v \times d^2 \end{aligned} \tag{4}$$

- The equation of consumed energy for the reception of $\mathcal{M}$ can be:

$$\mathcal{E}_r(\mathcal{M}_i^v)_{\mathcal{R}} = \mathcal{E}_{\mathcal{R}_{elec}}(\mathcal{M}_i^v) = \mathcal{E}_{elec} \times \mathcal{M}_i^v \tag{5}$$

where $\mathcal{E}_{\mathcal{T}_{trsm-elec}}$: transmitter circuitry(Energy dissipated by radio) , $\mathcal{E}_{\mathcal{T}_{trsm-amp}}$ is transmitter amplifier(dissipated energy for transmitting amplifier), $\mathcal{E}_{elec}$: Energy dissipated by transmitter or receiver circuitry per bit and $\mathcal{E}_{amp}$ is energy used to transmit amplifier per bit.

Upon these equations, the final $\mathcal{E}^{v}_{\mathbf{F}\ell}$ energy level of $\mathcal{V}$ sensor node can be expressed by equation (6):

$$\mathcal{E}^{v}_{\mathbf{F}\ell} = \left[\mathcal{E}^{v}_{init} - \left[\mathcal{E}_r(\mathcal{M}^{v}_{i}, d)_{\mathcal{T}_{trsm}} + \mathcal{E}_r(\mathcal{M}^{v}_{i})_{\mathcal{R}}\right]\right] \tag{6}$$

where $\mathcal{E}^{v}_{init}$ is initial energy of $\mathcal{V}$ sensor node.

The BS computes the position of sensor node $\mathcal{V}$ using the Euclidean positioning model in once the energy level calculation is completed. Given $\mathcal{A}$ region of deployment, $\mathcal{V}_x$ and $\mathcal{V}_y$ are the x and y positions of $\mathcal{V}$ node from BS. xBS and yBS are the x and y positions of BS. Then, the position of $\mathcal{V}$ sensor node to the BS can be expressed by equation (7)

$$\boxed{|\mathbf{D}[\mathcal{V}, BS]| = \sqrt{(\mathcal{V}x - xBS)^2 + (\mathcal{V}y - yBS)^2}} \tag{7}$$

After all, the BS rearranges each sensor node $\mathcal{V}$ in each cluster $\mathcal{K}$ based on the geographical positions $\mathbf{D}(\mathcal{V})$ of $\mathcal{V}$ sensor nodes from the BS and on their $\mathcal{E}_{\mathbf{F}\ell}(v)$ in the format expressed by equation (8):

$$BS \xrightarrow{\text{ordering}} \mathcal{N}(\mathcal{K}(v)) : [\mathcal{E}_r(\mathcal{V}), \mathbf{D}_r(\mathcal{V})] \tag{8a}$$

where

$$\mathbf{D}_r(\mathcal{V}) = \min(\mathbf{D}(\mathcal{V}, BS)) \tag{8b}$$

and

$$\mathcal{E}_r(\mathcal{V}) = \max(\mathcal{E}_{\mathbf{F}\ell}(\mathcal{V})) \tag{8c}$$

Thus, the $\mathcal{V}$ sensor node with highest $\mathcal{E}_r(\mathcal{V})$ energy level and has minimum position $\mathbf{D}_r(\mathcal{V})$ from BS elected as CH within each Cluster. Generally, BS qualified $\mathcal{V}$ as CH when it obeys the following properties:

$$\mathcal{R}ating(\mathcal{V}_i) > \mathcal{R}ating(\mathcal{V}_{i+1}) \tag{9}$$

and selects sensor node $\mathcal{V}$ as CH in each partitioned Cluster in the networks.

Right after, the BS generates a new message $\mathcal{M}$ consisting CLUSTER_HEAD_STATUS($\mathbf{CH}_{sts}$)

with two variables: *Leader* (CH) and *Member* (CM) as described in table 4:

TABLE 4. CH NODES STATES

Leader	Member	Status
1	0	CH
0	1	CM

The BS calculates a number of rounds the selected nodes can serve as CH role. Furthermore, the communication concerns cluster formation of nodes in this scheme is based on trapdoor signal $\mathcal{S}_\tau$ released by BS and current time information τ of sensor nodes. Thus, the BS encodes the message $\mathcal{M}$ consisting of consisting CLUSTER_HEAD_STATUS($\mathbf{CH}_{sts}$) before broadcasting to the sensor nodes by executing algorithm 4. Algorithm 4 takes $\mathcal{M}$, τ,$\mathcal{N}_{pk}$ and $\mathbf{CF}_{pk}$ of nodes as inputs and output encoded message $[\mathcal{M}_c]$

Algorithm 4 Algorithm of Cluster Formation Message Encoding

INPUT: N_{p_k}, τ, $\mathbf{CF}_{pk}$

function $CH_{sts}\_Message(\mathcal{N}_{pk}, \mathcal{M}, \mathbf{CF}_{pk}, \tau)$
  for $\mathcal{V} \in \mathcal{N}(\mathcal{N}(V))$ **do**
    Pick τ current time information
    Choose $\beta \in \overset{\mathcal{R}}{\leftarrow} \{0,1\}^n \in \mathbb{Z}_q$
    Compute $\mathcal{T} = \mathcal{H}_1(\tau)$ ▷ Hashing current time information where $t \in \{0,1\}^\tau$
    Compute $\alpha = \mathcal{H}_2(\mathcal{ID}_i^v \| \beta)$ ▷ where $\mathcal{ID} \in \{0,1\}^n$
    Compute $\omega = \alpha\mathcal{T}$
    Compute $\mathbf{CH}_{sts}$ codeword session $\Psi = \varphi(\mathcal{N}_{pk}, \omega) = \varphi(g^{\mathcal{N}_{sk}}, \alpha\mathcal{T}) = \varphi(g, \mathcal{T})^{\mathcal{N}_{sk}\alpha} \in \mathcal{G}_2$
    $[\mathcal{M}_1] = \alpha\mathbf{CF}_{pk} = \alpha g^\kappa$ ▷ where $\kappa \rightarrow \mathbf{CF}_{sk}$
    $[\mathcal{M}_2] = \mathcal{M} \oplus \mathcal{H}_3(\Psi) \in \{0,1\}^n$
  end for
  return $[\mathcal{M}_c] := \langle \mathcal{M}_1, \mathcal{M}_2, \tau, Status \in \{0,1\}, \mathcal{S}_\tau \rangle \rightarrow \mathcal{ID}_i^v$
end function

And finally the BS broadcasting $[\mathcal{M}_c]$ to each sensor node in with the following format:

$$BS \xrightarrow{[\mathcal{M}_c]} \mathcal{N}(\mathcal{V})$$

Broadcasting

However, the formation of clusters among sensor node is confirmed by **CF** computed in algorithm 3. Thus, all sensor nodes must pairwise and share its $\mathbf{CF}_{pk}$ with their cluster members $\mathcal{V} \rightarrow \mathcal{N}(\mathcal{K}(\mathcal{V}))$. The pairwise is conducted by all members of clusters until $\mathbf{CF}_{pk}$ shared by all Cluster Members. Then all sensor nodes update their routing table with Cluster Member information. In this case, one node can pairwise with only its Cluster Members.

After collecting the information, each sensor node must know their **CH** status. All sensor nodes should obtain trapdoor signal $\mathcal{S}_\tau$ executed in algorithm 2 by BS in order to know their **CH** status attached to $[\mathcal{M}_c]$, which is broadcasted from the BS. Thereby, each sensor node can know their **CH** status by executing algorithm 5. Algorithm 5 takes $\mathcal{M}_c$, N_{pk} , $\mathcal{S}_\tau$ and $\mathbf{CF}_{sk}$ of sensor node $\mathcal{V}$ and outputs $\mathcal{M}$ message containing elected CH in each Cluster.

Algorithm 5 Algorithm of CH Election Message decoding

INPUT: N_{pk}, $\mathcal{S}_\tau$, $\mathbf{CF}_{sk}$ and $\mathcal{M}_c$
OUTPUT: $\mathcal{M} \in \{\mathbb{M} \cup \perp\}$

function $\mathcal{M}_{ch\_decoding}(\mathcal{N}_{sk}, \mathbf{CF}_{sk}, \mathcal{S}_\tau, \mathcal{M}_c)$
  Given $[\mathcal{M}_c] := \langle \mathcal{M}_1, \mathcal{M}_2, \tau \rangle$,$\mathcal{S}_\tau$ and $\kappa = \mathcal{CF}_{sk}$
  for $\forall\ \mathcal{V} \in \mathcal{N}(\mathcal{K}(v))$ **do**
    if $\mathcal{S}_\tau \geq \tau$ **then**
      Compute $\mathcal{V}_i = \kappa^{-1} \times \mathcal{M}_1 = \kappa^{-1} \times \kappa\alpha g = \alpha g$
      Each node compute Session Codeword $\psi_i^v = \varphi(\mathcal{V}_i, \mathcal{S}_\tau) = \varphi(\alpha g, \mathcal{N}_{sk}\mathcal{T}) = \varphi(g, \mathcal{T})^{\mathcal{N}_{sk}\alpha} \in \mathcal{G}_2$
      $\mathcal{V}_i \leftarrow \mathcal{M}_{ch\_election} = \mathcal{H}_3\left(\psi_i^v\right) \oplus \mathcal{M}_2 \in \{0,1\}^n$
    else
      $\perp$
    end if
  end for
end function

Indeed, the correctness of obtained message $\mathcal{M}$ by $\mathcal{V}$ can be approved using equation (10).

$$\begin{aligned} \mathcal{V}_i \leftarrow \mathcal{M}_{ch\_election} &= \mathcal{H}_3\left(\psi_i^v\right) \oplus \mathcal{M}_2 \\ \mathcal{V}_i \leftarrow \mathcal{M}_{ch\_election} &= \left[\mathcal{H}_3\left(\psi_i^v\right) \otimes (\mathcal{M} \otimes \mathcal{H}_3(\Psi))\right] \\ \mathcal{V}_i \leftarrow \mathcal{M}_{ch\_election} &= \left[\mathcal{H}_3(g, \mathcal{T})^{\alpha\mathcal{N}_{sk}} \otimes (\mathcal{M} \otimes \mathcal{H}_3(g, \mathcal{T}))^{\alpha\mathcal{N}_{sk}}\right] \\ \mathcal{V}_i \leftarrow \mathcal{M}_{ch\_election} &= [\mathcal{M}] \end{aligned} \tag{10}$$

The $\mathcal{M}$ message consists two variable values: **Leader** and **Member** which determine the Cluster Head and Cluster Member of $\mathcal{V}$ node respectively. Meanwhile, $\mathcal{V}$ node decoding $\mathcal{M}$ message, if the values of ***Leader*** and ***Member*** are **1** and **0** respectively, $\mathcal{V}$ node is Cluster Head (CH). Otherwise,$\mathcal{V}$ node is Cluster Member (CM) as stated in table 4. The $\mathcal{V}$ node elected as CH broadcast its *Cluster_Head_Status* to each sensor node belongs to its cluster. Finally, it receives the join request from CM and starts for aggregating sensed data from BS.

5.3.1.4. Data Collections

After the trustworthy election of CH, the elected CH collects the sensed data from the cluster members and transmits to the BS in its assigned time slot. Then, ready for the next CH data aggregation starts again.

5.3.1.5. CH Role Rotation

The proposed scheme also computes the CH role rotation whenever the energy level Cluster Head (CH) is beyond an assumed threshold value. We consider a CH role rotation while its lifetime has run out of energy at current round. We use energy model adopted in [70] to calculate the energy utilization rates of Cluster Head (CH) and Cluster Member (CM) per each cluster for the sake of CH role rotation.

Given there are $n+1$ sensor nodes in the cluster such that the energy consumption of CH expressed as in equation (11).

$$\mathbf{E}_{ch} = \mathbf{E}_{sense} + n\left(\mathbf{E}_{Rx} + \mathbf{E}_{process}\right) + \mathbf{E}_{Tx}(\mathcal{C}) \tag{11}$$

Energy utilized by Cluster Member (CM) can be expressed as on equation (12).

$$\mathbf{E}_{cm} = \mathbf{E}_{sense} + c\mathcal{E}_{Tx} \tag{12}$$

Put all these together, the proposed scheme design TRA to detect node replica attacks as follows:

Algorithm 6 Detection Scheme Algorithm

INPUT: sensor node $\mathcal{V}, \mathcal{E}_r, \mathcal{P}_{stn}$

function CLUSTER_ FORMATION($\mathcal{V}, \mathcal{E}_r^{(v)}, \mathcal{P}_{stn}^{(v)}$)
  if *cluster.Status = Zero* **then**
    for $\forall$ *node* $\mathcal{V} \in \mathcal{N}(v)$ **do**
      Construct $\mathbf{CF}^v = \mathbf{CF}_1^v \bigvee \mathbf{CF}_2^v \bigvee \mathbf{CF}_3^v \bigvee \ldots \bigvee \mathbf{CF}_{\mathcal{N}}^v$ ▷ See algorithm 3
      $\mathcal{BS} \Longleftarrow \mathcal{V} = \mathcal{B}r_{oadcast}\left(\mathbf{CF}_{\mathcal{N}}^v, \mathcal{E}_r^{(v)}, \mathcal{P}_{stn}^{(v)}, \mathcal{ID}_{\mathcal{N}}^v\right)$
      if $existing\left[\mathcal{ID}_{\mathcal{N}}^{\mathcal{V}_i}, \mathbf{CF}_{\mathcal{N}}^{\mathcal{V}_i}\right] = received\left[\mathcal{ID}_{\mathcal{N}}^{\mathcal{V}_{i+1}}, \mathbf{CF}_{\mathcal{N}}^{\mathcal{V}_{i+1}}\right]$ **then**
        if $\mathcal{P}_{stn}^{(v)_i} \neq \mathcal{P}_{stn}^{(v)_{i+1}}$ **then**
          $clone \;\; detected\mathcal{V}^{'}$
        end if
      else
        for $\forall$ $\mathcal{V} \in \mathcal{N}(v)$ **do**
          $\mathcal{BS} \downarrow\uparrow \mathcal{V}_i \in [\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \ldots, \mathcal{V}_{\mathcal{N}}) : \mathcal{P}_{stn}^v$ ▷ $\mathcal{P}_{stn}^v$ based $\mathcal{V}$ ordering
          if $\mathcal{V}_i$ *and* $\mathcal{V}_{i+1} < \mathbb{C}_{rng}$ **then**
            Assign $\mathcal{V}_i$ and $\mathcal{V}_{i+1} \rightarrow \mathcal{K}_\iota$
          else
            Compute new $\mathcal{K}_\iota$ and Increment cluster $\mathcal{K}_{i+1}$
          end if
          Increment node $\mathcal{V}_{i+1}$
          $\mathcal{SH}_{are}$ $\mathbf{CF} : \mathcal{V}_i \rightarrow \mathcal{K}(v)$
        end for
      end if
    end for
    for $\forall$ *node* $\mathcal{V} \in \mathcal{K}(v)$ **do**
      Rank node $\mathcal{V}_i$: $\mathcal{BS} \rightarrow \mathcal{V}_i :\rightarrow (\mathcal{E}_r^v, \mathcal{P}_{stn}^v)$
      if (RANK($\mathcal{V}_i$) > RANK($\mathcal{V}_{i+1}$)) ($\mathcal{V}_i \neq \mathcal{V}_{i+1}$) **then**
        $\mathcal{V}_i = \mathcal{CH}$
        Add node $\mathcal{CH}$ to $\mathbf{CH}_{lists}$
      end if
      $\mathcal{V}_{i+1}$
    end for
    for $\mathcal{CH} \in \mathbf{CH}_{lists}$ **do**
      Rank $\mathcal{CH}_{lists} :\rightarrow \{\mathcal{P}_{stn}^v\}$
      Calculate $\mathcal{R}_n :\rightarrow \mathcal{CH}_i$ ▷ Round of CH role of $\mathcal{V}_i$
    end for
    for every $\mathcal{V}_i \in \mathcal{K}_i$ **do**
      $\mathcal{BS} \rightarrow \mathcal{K}_i(\mathcal{V}) : \{\mathcal{CH}_i\}$
      $i + 1$
    end for

```
39:         for every $\mathcal{V}_i \in \mathcal{K}_i$ do
40:             $\mathcal{CH}$ broadcasts its $\mathcal{CH}_{adv}$ status to Cluster Members
41:             if ($\mathcal{S}_\tau \geq \tau$) and (Leader == 1) then
42:                 $\mathcal{CH}$ = TRUE
43:                 $\mathcal{V}_i$ receive CH role $adv$
44:             else
45:                 cloned $\mathcal{CH}'$ detected
46:             end if
47:             if (Member == 1) and ($\mathcal{CH}_{adv\_rcvd} == 0$) then
48:                 node $\mathcal{V}$ send Join Request to CH
49:                 if $\mathbf{CF}^v$ $\exists$ $\mathcal{CH}_{member\_lists}$ then
50:                     $\mathcal{V}$ joins Cluster Head (CH)
51:                 else
52:                     node replica $\mathcal{V}'$ detected
53:                 end if
54:             end if
55:         end for
56:     end if
57: end function
```

5.3.2. Security Analysis

5.3.2.1. Replica attack Detection

Our detection scheme is more resilient against node replica attacks being able to participate as legitimate in cluster formation operations of the networks. If legitimate node $\mathcal{V}$ captured and compromised by an adversary, it is easy for an adversary to compromise all the credentials belongs to that node. If we expect that an adversary requires to proceeding the CH election process before the release time, the node replica can be identified by cluster members while cross-checking $\tau > \mathcal{S}_\tau$

As stated, in section 5.2.1.1, all sensor nodes are static after deployment and after node clustering process also completed. It means that $\mathcal{V}$ node can be a member of a fixed cluster, whose **CF** encoded from its Cluster Members and shared with only its CMs. In this detection scheme, a message concerns any clustering process broadcasted into the future. Therefore, whenever the BS broadcasts clustering messages to $\mathcal{V}$ nodes, its signed with the current time information τ , **CF** and Network codeword of $\mathcal{V}$ node as stated in algorithm 4 line 12. We consider, each sensor node $\mathcal{V}$ computes **CF** with only their Cluster Members. Assume node replica may compute new clustering message $\mathcal{M}_2'$ and sign it by using **CF** and Network codeword of a legitimate node. Moreover, we evaluate the effect of cloned $\mathcal{V}'$ nodes on the detection algorithm in cluster formation process based on the following parameters:

5.3.2.2. Detection during CH Candidates Screening

Before deployment of the sensor nodes, the BS maintains the indexing of ϕ_{params} assigned to node $\mathcal{V}$ with its $\mathcal{ID}s$. After clustering nodes into groups, each sensor node $\mathcal{V}$ in the clusters can insert itself in the CH candidate $\mathbf{CH}_{cand.}$ screening process for being elected as a CH. After BS receives a $\mathbf{CH}_{cand.}$ message $\mathcal{M}_{CH\_cand.}$ from node $\mathcal{V}$, it verifies the trust level of that message by cross-checking whether codewords encoded in the message found in the record of the BS. If $\mathcal{V}$ node codewords match with the other nodes(the same with $\mathcal{ID}$ of another node) or not found in the record, it indicates that there must a node replica attacks in the networks. The BS screen out node replica attacks from CH candidates and proceeds the CH election process among legitimate sensor nodes.

5.3.2.3. Detection during CH role Advertising

We need to show that any polynomial time τ of the BS, sensor node $\mathcal{V}$ does not output $\perp$ and $\tau \geq \mathcal{S}_\tau$ then the sensor nodes can compute $\mathcal{M}_2$ consisting of CH status broadcasted from the BS. In other words, the detection scheme guarantees that the sensor node $\mathcal{V}$ can be able to receive CH status messages $\mathcal{M}_2$ if and only if the current time τ of the node $\mathcal{V}$ was not earlier than trapdoor time signal $\mathcal{S}_\tau$.

Suppose $\mathcal{V}$ nodes are compromised by the adversary after deployment and clone the compromised nodes $\mathcal{V}^{'}$ and deploy the clone nodes after node clustering and before CH elections in different locations of Clusters $\mathcal{K}$. Then, we can say that $\mathcal{V}^{'} \in \{\mathcal{V}_1^{'}, \mathcal{V}_2^{'}, \mathcal{V}_3^{'}, \ldots, \mathcal{V}_n^{'}\}$ can participate in cluster formation as a legitimate node.

Assume node replica $\mathcal{V}^{'}$ computes n bits of message $\mathcal{M}^{'}$ consisting CH status and advertises a request that its a Cluster Head to its cluster members before trapdoor $\mathcal{S}_\tau$ is released. It means that the node replica broadcasts its Cluster Head in polynomial time $\tau > \mathcal{S}_\tau$. After receiving a CH request from node replica $\mathcal{V}^{'}$, each cluster members computes session codeword $\mathcal{K}$ as stated in algorithm 5 on line 6 and cluster members verify the consistency of the message as expressed in equation (10).

Since the message is broadcasted before trapdoor time signal, it returns $\perp$. Then, sensor node $\mathcal{V}$ triggers an alarm to the BS. The BS sends the query to networks and revokes clone $\mathbf{ch}^{'}$ from the networks.

5.3.2.4. Detection during Joining Assigned CH

When the BS assigns CH to each cluster, before CH message has been broadcasted, it identifies the list of all members belongs to the fixed cluster and attach cluster members file to the elected CH. After node $\mathcal{V}$ has been elected as Cluster Head , it broadcasts the message that contains its Cluster Head to Cluster members. Suppose a sensor node $\mathcal{V}$ compute new n bits message m consisting $\mathbf{CH\_}\mathcal{JOIN}$ and send to a CH. When a CH

receives a **CH_$\mathcal{JOIN}$** message m , it checks for the node $\mathcal{V}$ has a consistent **CF** match with its record and in its Cluster Members list. If a node $\mathcal{V}$ does not belong to that cluster or **CF** of node $\mathcal{V}$ identified for a mismatch, a CH must raise an alarm to the BS. Then the BS revokes node replica attacks from the networks afterward.

5.3.2.5. Detection during CH Role Rotation

When node $\mathcal{V}$ elected as BS, the BS calculates the number of rounds (time slots divided into CH election and data transmission/aggregation phases) node $\mathcal{V}$ serves as a CH role before broadcasting CH message and announcements. Thus, when energy level $\mathcal{E}_{\mathbf{F}\ell}$ of node $\mathcal{V}$ holds the current CH role below the energy threshold $\mathcal{T}(e)$, the BS raise CH candidates among sensor nodes in the clusters as depicted . The BS removes node replica attacks from CH candidates with the same steps in section 5.3.2.2 and rotate the role CH in the legitimate node $\mathcal{V}$.

Algorithm 7 Algorithm of Secured CH Role Rotation

INPUT: : $\mathcal{CH}$

```
1: function CH_Role_Rotation(CH)
2:     for each CH_i ∈ K_i(CH) do
3:         if R_n == TRUE then
4:             Make current CH_i sessions idle
5:             BS ← CH_i Role Rotation Status
6:         else
7:             Retain CH_i as it is CH role for next round
8:         end if
9:     end for
10: end function
```

6. Simulation and Result Evaluation

6.1. Overview

In this section, several simulation scenarios are described and results found from simulations are analyzed and presented. In section 6.2, simulation tool and framework modules are discussed and introduced. Section 6.4.1 present and discusses the most important network parameters used in the simulations. In section 6.4, the results obtained from simulations are discussed and presented. Finally, the comparison between the proposed scheme and the existent scheme against node replica attacks will be presented.

6.2. Simulation Tool

Setting up a network to do some real experiments is the best way for studying about communication on the Internet. However, due to cost and complexity, setting a network is not easy. Thus, Network simulator is a tool that provides a virtual network for network simulation in only one computer. Network simulation is the most known methodology in the world of network used to implement and evaluate different network scenarios before real-world implementation. There are number network simulators such as WSNet, COOJA, NS-2,NS-3, OMNET++,J-SIM,NetSim, etc. All these tools are applicable to WSNs simulation. However, selecting a well tested and an appropriate simulator is suitable to obtain optimal performance results and implementing network scenario in a better manner. OMNET++ is the most tested and popular network simulator all over the world to simulate wired as well as wireless network topologies. [71].

OMNET++ follows a hierarchical component-based architecture for modeling a simulation scenario. Models are assembled from OMNET++ reusable components, called modules. In OMNET++, modules communicate with each other by exchanging event message. These modules can be a *Simple module* , *Compound module* ,and Network module. A Simple module is the lowest level module and programmed using C++ library. The Simple modules are grouped hierarchically into compound modules. The compound module is the module that consists varying hierarchical simple modules and also sometimes other inherited compound modules. Network module is a top-level module that comprised compound modules and simple modules. It represents a complete OMNET++ simulation model. The messages exchanged between modules represents the events. Messages can be transmitted from one simple module to another via either directly to the modules or via *gates* interface and connections (the linking between an output gate and input gate). Gates represents the input and output interfaces of modules. Modules use this interface to send messages to another module or directly without the gate. The gates facilitate three media communication

interfaces: input to receive a message, output to send the message and inout interface that can serve as input also output.

In OMNET++, modules (including simple, compound module and network), gates and connections are described by OMNET++'s language Network Definition Language (NED). The NED follows object-oriented features like inheritance and interfaces. It also supports Java-like structure to reduce the conflict between the module's name.

6.3. The Simulation Models Components

This section describes varying modules implemented and how they are assembled to form the simulation model used in this scheme. It also explains how the modules were configured to generate the desired results. The simulated network scenario contains five module components: Network module, Sensornode module, Basestation module, Clone attack module and key-management module. The implementation details of these modules presented in Appendix 7.

- **Torching module** In this scheme, the `Torching` module is considered as a network module that consists the entire modules needed to simulate the proposed scheme. It extends WSN module that provides a basic parameters of sensornodes. `Torching` defines vector sensornodes and several parameters of the simulated network.
- **The SensorNode module** This module the blueprint module of this scheme, named `SensorNode` which used in different simulation modules of this simulation. This module consists it's own simulation parameters and gates to communicate with other modules. Specifically, it monitors and aggregate event's data from the deployment area and forwarded to Basestation module. Meanwhile, it also notifies if clone attack behavior has been detected. See appendix 7.
- **Basestation module** The Basestation module is the module created to control the entire events of the simulation. It also facilitates the control service to monitor the communication between modules. Specifically, it receives all information forwarded from Sensornode module and provide security between modules. Also, it is designed to detect and mitigate clone attacks.
- **Clone attack module** For clone attack simulation, this module is created in this network scenario to facilitate the characteristics of clone attack. It inherits various parameters from the `WSN attack` module. It also contains its own gates and parameters that represent its definition. However, in this simulation scenario, this module is created dynamically.
- **The Key-management module** `KeyManagement` is an independent module that provides the pair-wise and security functions for a sensornode module.

This module deliver it's service by embedded to sensor-node module.

6.4. Evaluation metrics and Result Analysis

We have simulated the proposed clone detection algorithm and compared the evaluated results with an exist clone detection scheme [66] since both schemes are suitable for detect clone attack in cluster formation world. We implemented our network with cluster communication model as discussed below. For our experiments, we have deployed two different network scenarios. Here, we have detailed the scenarios.

6.4.1. Simulation Parameters

We simulated the proposed scheme in OMNET++ simulator. We evaluated the proposed scheme with several simulation parameters like network size, random deployment, energy distribution and other related issues.

As depicted in chapter 4, sensor nodes are deployed in a random fashion with the limited energy level in their batteries. In the defined simulation scenario, the sensor nodes are considered as stationary and randomly distributed in the area within 2000cm × 1000cm ; assumed only one sink node in the simulation scenario. The sink node is assumed to have sufficient unlimited energy, which enables it to stay until the end of events and end of the network. As a whole, Table 5 illustrates the considered as simulation parameters for the simulated network.

TABLE 5. Simulation Parameters

	Parameter	**Value**
Simulation Parameters	Population of SensorNodes	50
	Width of Simulation area	2000m
	Length of Simulation area	1000m
	Deployment fashion	uniformly random
	Communication range	100 m
	Simulation time	15m
	Clone attack deployment	Dynamic
	Sensor Node Energy	uniformly random
	Message length	512B
	Xpos	Random
	Ypos	Random
	Topology Type	Cluster

Scenario 1 : We have simulated this network scenario to evaluate the sustainability of the proposed scheme to process optimal CH election and cluster formation. In this scenario, sensornodes were arranged to form a cluster topology based on the proposed scheme. This network scenario was configured this without clone attack.

As discussed in the previous chapter, prior to deployment each sensornode preloaded with a security mechanism proposed in this scheme. Afterward, each sensornode deployed randomly to network area as depicted in figure 23. Next deployment, each sensornode

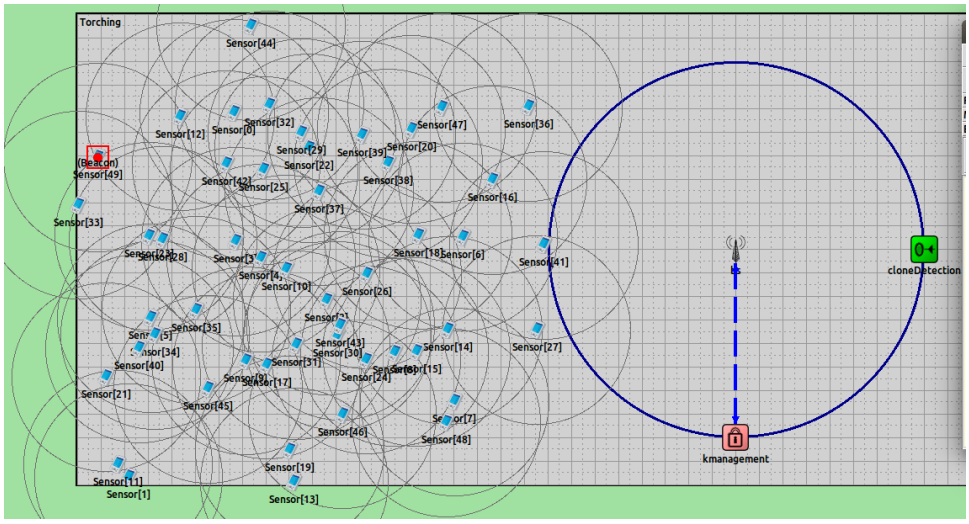

Figure 23. Sensornode deployment

sends the status packet to the Basestation module that needed for cluster formation between sensornode module. In this case, the status packet consists all parameters required for cluster formation and a bit parameters of Basetstation as shown in figure 24. After receiving

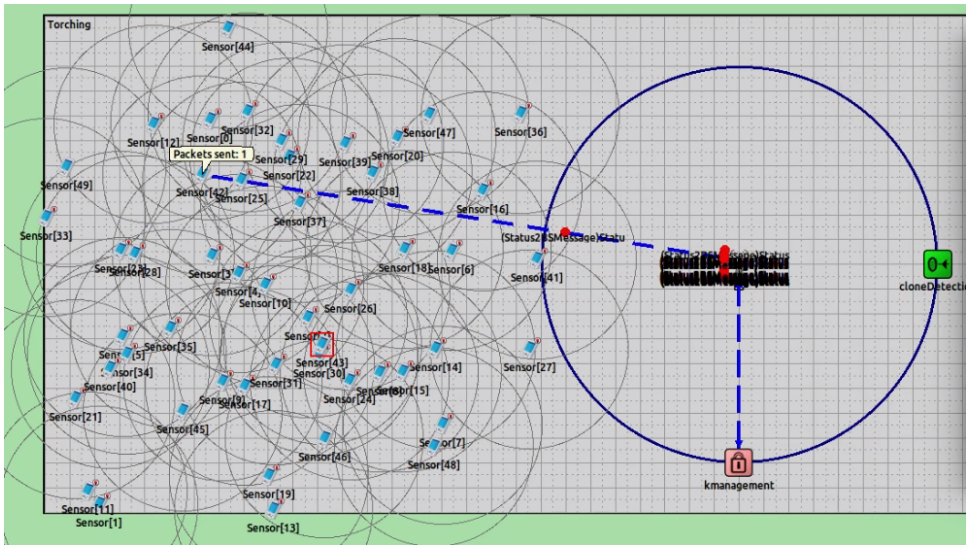

Figure 24. Status transmission

the status message, the Basestation computes the cluster formation based on the message forwarded from sensornodes. In other words, the Basestation organizes sensornodes in the cluster and then the sensornode that fulfills the requirements of the proposed scheme has been elected CH and assigned to each cluster. As discussed in the previous chapter, the cluster formation is formulated based on the proposed algorithm to formulate optimal clustering. The cluster formation has been formulated as shown in figure 25. Finally, the sensornode aggregates the data and forwarded to the Basestation.

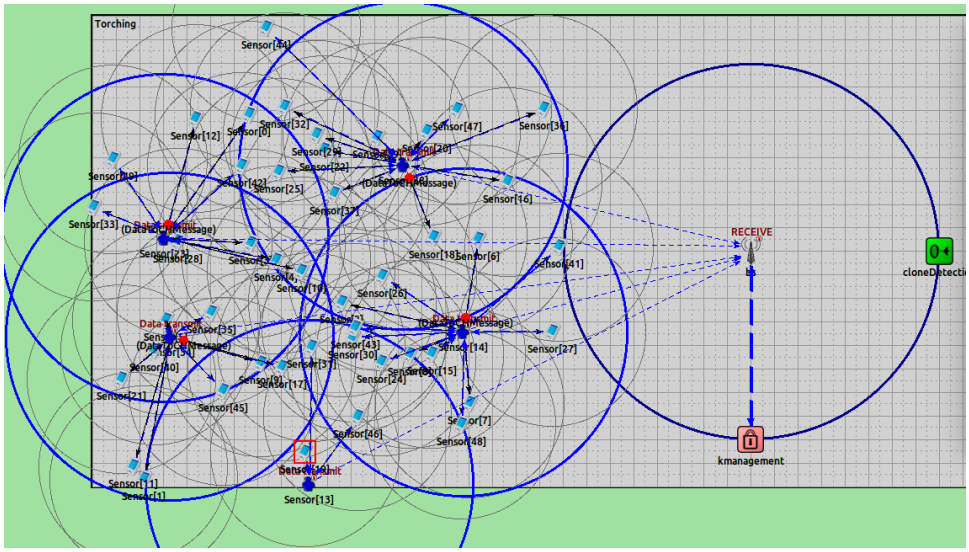

Figure 25. Cluster formation and Data transmission

Scenario 2 : In this scenario we have implemented two types of network simulation: Normal Network Scenario (NNS) and Compromised Network Scenario when clone attack active (CNS). In the CNS, the network simulation was deployed with clone attacks. We have simulated a network with 50 sensornodes plus the number of detected clone attacks as shown in figure 26. Afterward, we have analyzed the result for both scenarios in case

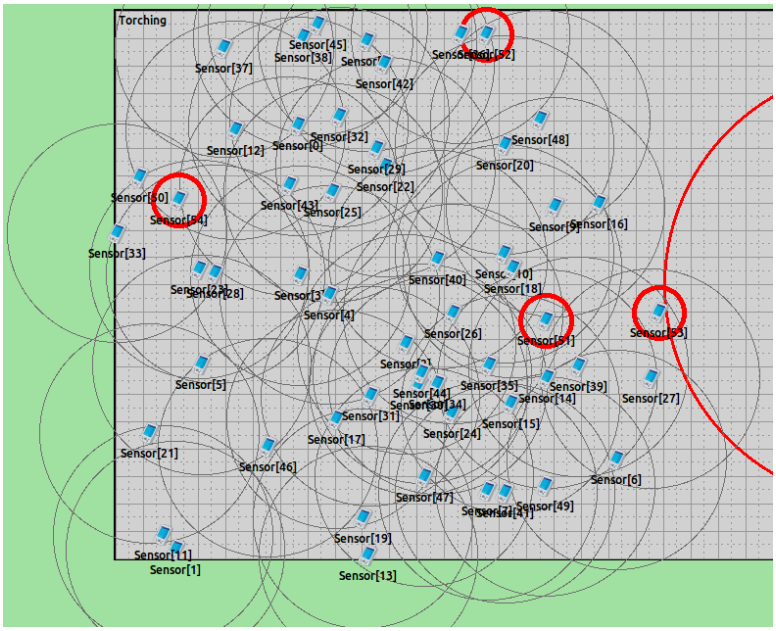

Figure 26. WSN network with encircled clone attacks

of total energy consumed, total number of packets sent and received(communication cost) and Network lifetime/dead nodes with varying network size.

The simulation results for both scenarios have been discussed as follows:

6.4.1.1. Communication cost

The communication cost investigated in this section is the cost associated with the total number of packets sent and received during network operations. Fig 27 shows the comparison between the results found based on the average number of packets sent/received in NNS and CNS simulations. As observed from Fig 27, the number of packets sent/received

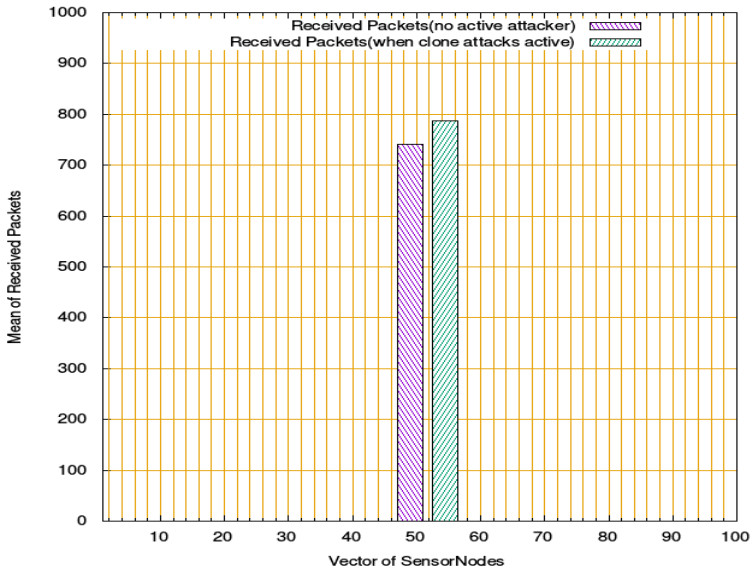

Figure 27. Total Number of Received Packets

during CNS operation is higher than NNS operation. This is due to the fact that clone attack acts as legitimate nodes to neighbors to operate the WSN tasks and consequently increase the volume of sent/received packets. This leads to higher network packets around the CNS. In the other word, due to the arrival of unexpected data packets from clone attacks, the behavior of packets transmitted in the CNS might deviate from average packets operated in NNS. This influences prompting the sensornodes to send/receive a higher number of packets. This feature can be used for the detection of clone attacks.

The same simulation also was done for both network scenarios with different network size, which resulted in the same outcome. The clone nodes were detectable as shown in Fig 26.

6.4.1.2. Energy consumption

The total energy consumed for both network scenarios is presented in Fig. 28. In both simulations, each sensornode periodically calculates the amount of energy they consumed. The predicted energy consumption rate is calculated based on the average number of sent/received packets by each sensornode. Comparing the amount of energy consumed in NNS with the predicted energy consumption rate in the CNS, sensornodes in CNS consume more energy than the sensornode in the NNS scenario because of the clone attack route the data packets over neighbor nodes. It indicated that the sensornodes in CNS process the

data packets exceed thresholds since the prediction of the energy consumption rate is high in comparison with NNS. This proves that the higher the number of packets sent/received , the higher the consumption of energy map construction. Moreover, if transmitted data packets exceed the threshold value, the power of CH and sensornode will quickly deplete and makes cluster members may unable to send the information to the BS. Therefore, CNS has a high impact on the lifespan of WSNs applications. Fig. 29 also depicts that

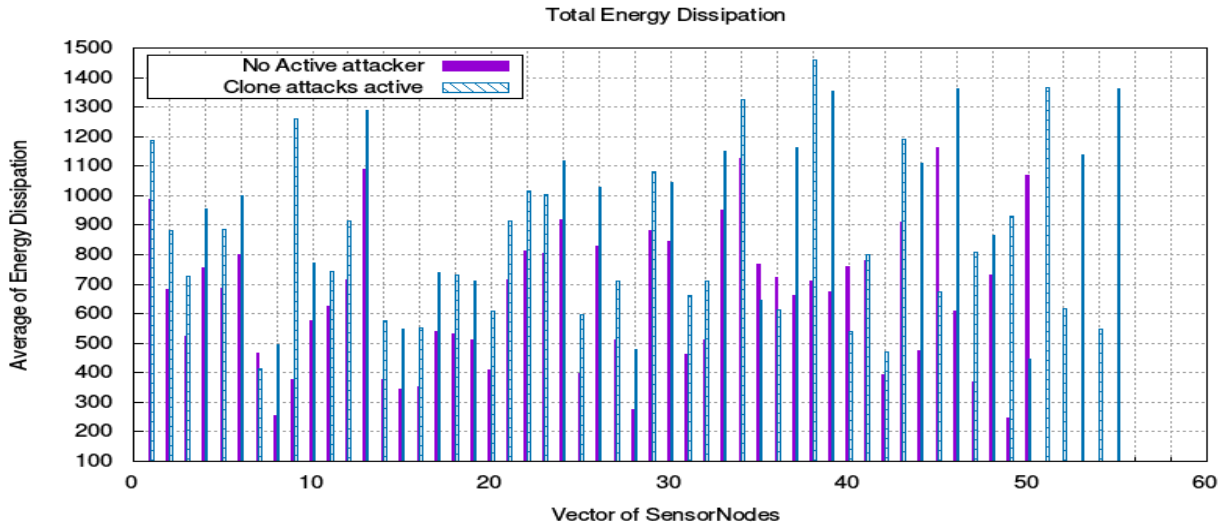

Figure 28. Total Energy Consumption per node

total energy consumption per rounds for both network scenarios. It demonstrated that as the number of rounds increased the packets sent/received generally increases, which consequently increases the total energy consumed. The results of the simulations performed also showed that high energy consumed per each round in the CNS in comparing with NNS.

As a result, the nodes cannot remain alive for more rounds and forward the information to the BS and consequently, which shortens the lifespan of WSNs applications.

6.4.1.3. Network lifetime

Fig 30 presents the comparison of simulation results of clone attack effects on the lifespan of Wireless Sensor Network for both scenarios. Here, we have evaluated the effect of both scenarios on the lifetime of WSNs. From the graph, it could be seen that the number of dead nodes in the CNS is higher than NNS in each round. Moreover, when the number of dead nodes increases with an increase in the number network round operations. The simulation result is observed until the end of the simulation (i.e. The power of the

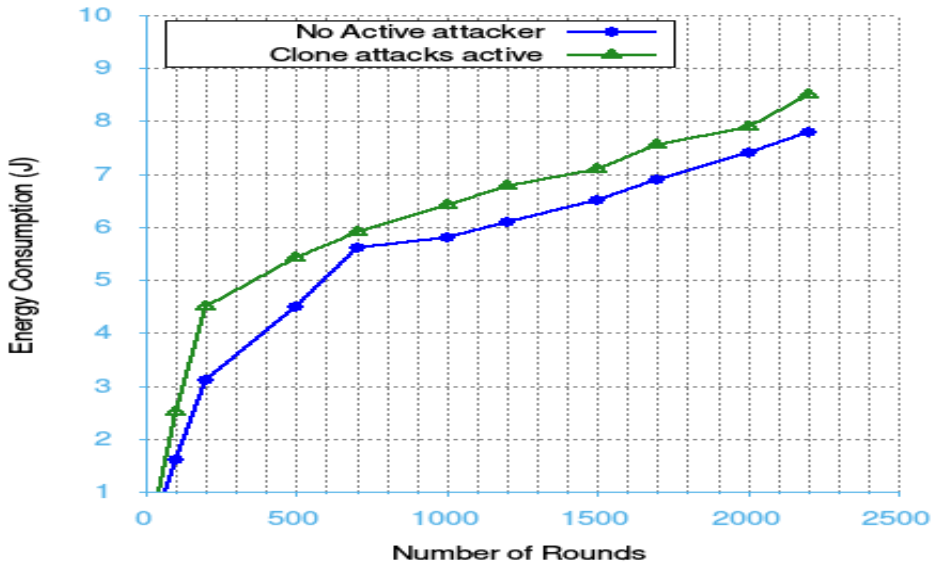

Figure 29. Total Energy consumption per round

sensor node is zero). The reason is that the packet broadcasted with clone nodes influences the energy of sensornodes for the demands of WSNs operation is based on the energy of sensornodes. In other words, when the rate packet transmission increases, the lifetime of the network will decrease. As a result, the alive of clone attacks in the network will affect the lifespan of WSNs applications.

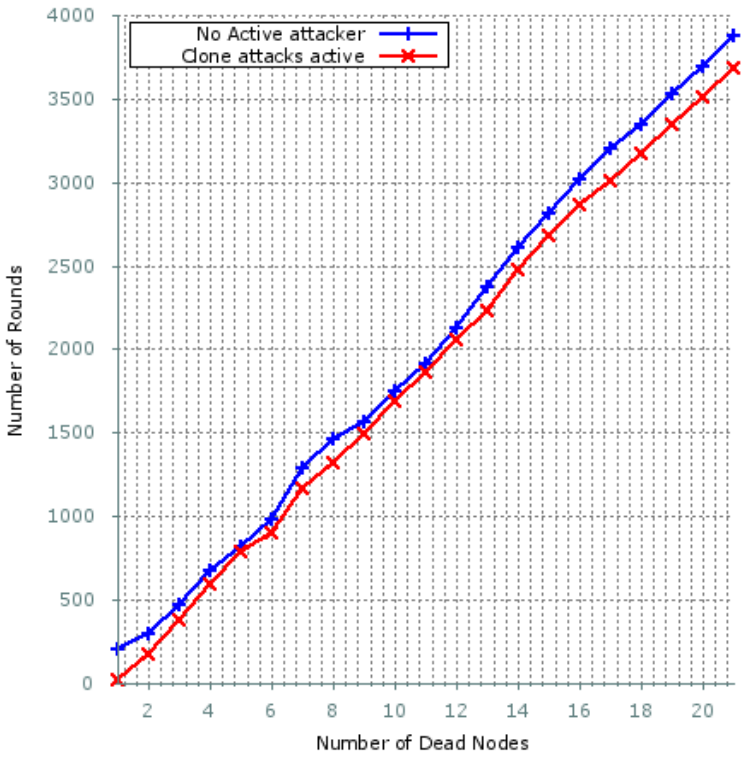

Figure 30. Lifetime of network Operation

6.5. Comparing the Proposed Scheme with ZBNRD Protocol

This section provides the comparison of the results of the proposed scheme and existent node replica detection scheme (ZBNRD [66]) which is more related to the proposed algorithm. Here, we have analyzed the communication cost, total energy consumption to detect a node replica attacks in both schemes. Also, we have compared the results in for both schemes in case of a number of cloned sensornodes.

6.5.1. Communication cost

In Figure 31, the communication cost comparison of both schemes has been presented. In this case, we investigated the communication overhead of the proposed scheme and ZBNRD protocol in case of clone attack detection. ZBNRD protocol makes each CH transmit $r(n^2 - n)$ packets during the operation of clone attack detection; where r is the number of detection rounds and n is the number of CHs in the network. On the other hand, if there is n CHs in the networks, one CH receives $n - 1$ packets from another CHs to detect a clone attacks. As a result, when network size increase, the broadcasted detection messages will be increases which makes high communication overhead. Consequently, it convinced that ZBNRD protocol is not suitable to detect a clone attack in dense network size. As depicted in Fig 31, our scheme greatly reduces the packet volume required to detect a clone attacks. Because our scheme significantly limited the operation of clone attack detection to intra-cluster, no packet forwarded to the neighbor CHs during detection session. Significantly, our scheme reduces the communication overhead

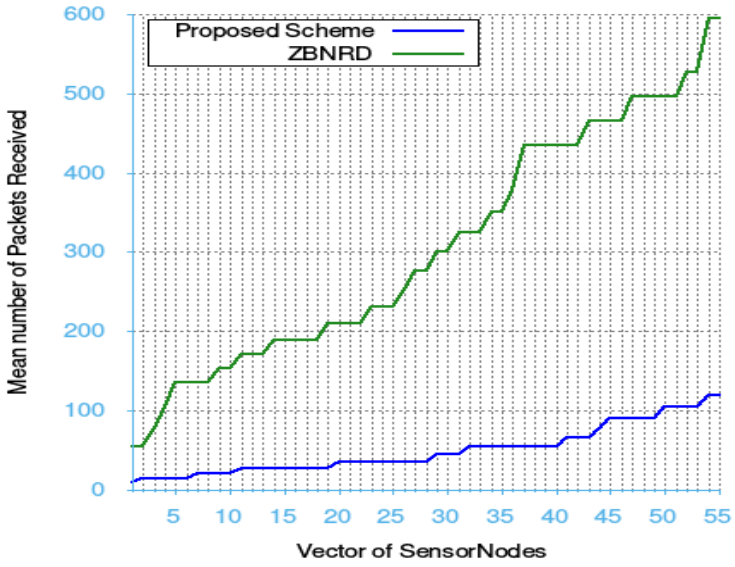

Figure 31. Communication Cost

6.5.2. Total energy consumption

To increase network lifetime, sensornode's energy must be saved when designing and developing WSN applications. Therefore, it is very important to reduce energy consumption while maintaining WSN applications for the sake of extending the network's lifetime. Fig 32 presents the average energy consumed to detect a clone attack for both protocols. This figure proves that the ZBNRD[66] consume more energy than the proposed scheme. The ZBNRD protocol uses inter-cluster mechanism to detect a clone attack which requires a large number of a packet being generated. As depicted in Figure 32 the proposed scheme consumes less energy to detect a clone attack as compared to the ZBNRD protocol. The reason less energy consumption in the proposed scheme in comparison to ZBNRD protocol is because the proposed scheme detects a clone attack based on the trapdoor and **CF** community member detector mechanisms. These two mechanisms are limited clone attack detection to intra-cluster. Instead of forwarding the requested **REG_MEMB** message to the entire cluster community like ZBNRD protocol, the proposed scheme identified the exact cluster community and open the cluster community door for sensornode using **CF** community detector and trapdoor mechanisms respectively. Moreover, the proposed scheme has no explicit exchange of detection messages with neighbor clusters to detect cloned attacks . As a result, the proposed scheme achieved a better performance when compared with ZBNRD protocol.

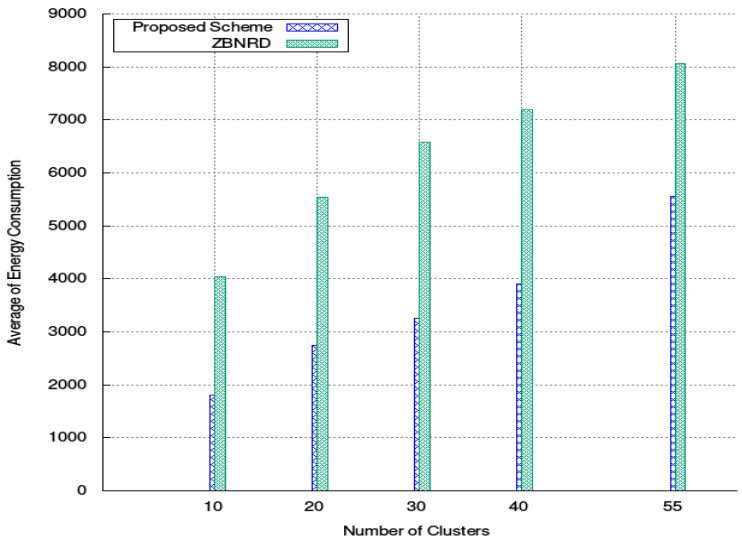

Figure 32. Total Energy Consumption to detect clone attacks

6.5.3. Number of Detectable Clones

Fig 33 presents the number of detected clone attacks during the CH selection process. As depicted in Fig. 33, the number of detectable clones increases with an increase in the size of the networks. Although the increase in the network size causes more communication

overhead, the cloned nodes can hide the CH election process in ZBNRD. Because with an increase in the size of the network, the number of the cluster will increase. Therefore, the cloned nodes can be elected as CH and can keep aggregating data from the cluster members for a long time in ZBNRD scheme. That makes the difference of detecting cloned nodes that analyzed in both schemes is quite big as shown in Fig. 33.

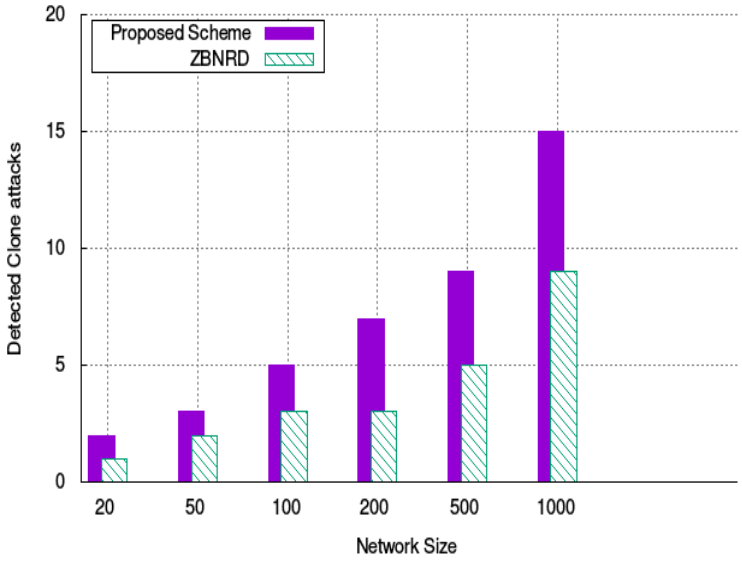

Figure 33. Number of Clones Detected

7. Conclusion and Future Work

In this paper, we presented a node replica detection scheme which greatly detects and mitigates a negative impact of cloned CH on the network. On purposely, the proposed scheme promising two techniques: First, at the first round of CH election, the proposed schemes evaluates the trustworthiness of each sensor node and expel out some node replica from CH candidates. Second, during CH role rotation, the proposed scheme assigns a CH role to the legitimate node. By doing these, the proposed scheme exposed the data gathered by clone CH'to the adversary which significantly extends the lifespan of WSNs. We evaluated the performance security of the proposed scheme and existent scheme through analyses. The proposed scheme exceeds over the existing scheme in terms of detection probability, communication cost, network lifetime and energy consumption. Moreover, the above analysis showed that the proposed scheme causes much less overhead than the existing scheme. As a future work, we plan to design and develop sinkhole attack detection in WSNs using a well-known simulator for proving the security and performance of our schemes over existent schemes.

References

[1] I. F. Akyildiz, *Wireless Sensor Networks*, 1st ed. A John Wiley and Sons, Ltd, Publication, 2010.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks(ELSEVIER)*, vol. 52, pp. 2292–2330, 2008.

[3] Y. W. Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 1–23, 2006.

[4] S. Kaplantzis, a. Shilton, N. Mani, and Y. Sekercioglu, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks using Support Vector Machines," *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, 2007.

[5] S. Lmdsey and C. S. Raghavendra, "PEGASIS : Power-Efficient GAthering in Sensor Information Systems," in *IEEE Aerospace Conference*, 2002, pp. 1125–1130.

[6] H. Jadidoleslamy, "A HIERARCHICAL I NTRUSION D ETECTION ARCHITECTURE FOR WIRELESS SENSOR NETWORKS," *International Journal of Network Security & Its Applications*, vol. 3, no. 5, pp. 131–154, 2011.

[7] P. Tran, A. Quang, and D.-s. Kim, "Clustering Algorithm of Hierarchical Structures in Large-Scale Wireless Sensor and Actuator Networks," *Journal of Communications and Networks*, no. October, 2013.

[8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks : a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[9] G. J. Pottie, "Wireless Sensor Networks," *IEEE Information Theory Workshop*, pp. 139–140, 1998.

[10] M. M. N. Aldeer, "A Summary Survey on Recent Applications of Wireless Sensor Networks," *IEEE Student Conference on Research and Development (SCOReD*, no. December, pp. 16–17, 2013.

[11] T. Arampatzis, J. Lygeros, and S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," *Proceedings of the 2005 IEEE International Symposium on , Mediterranean Conference on Control and Automation Intelligent Control*, pp. 719–724, 2005.

[12] C. Townsend and S. Arms, "Wireless Sensor Networks:," in *Sensor Technology Handbook*, 2004, pp. 439–450.

[13] J. S. A., "Wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1370–1379, 2006.

[14] D. Estrin, J. Heidemann, S. Kumar, and M. Rey, "Next Century Challenges : Scalable Coordination in Sensor Networks," *Proceedings of the ACM MobiCom'99, Washington, USA,*, no. Section 4, pp. 271–278, 1999.

[15] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, J. Zhao, M. Rey, and J. Zhao, "Habitat Monitoring : Application Driver for Wireless Communications Technology," *SIGCOMM Comput. Commun. Rev. 31(2 supplement),*, pp. 20–41, 2001.

[16] F. A. Silva, P. Gerhard, M. C. R. Talampas, and D. Pesch, "Industrial Wireless Sensor Networks: Applications, Protocols, and Standards," *IEEE INDUSTRIAL ELECTRONICS MAGAZINE*, no. december, pp. 67–68, 2014.

[17] M. Jan, M. Josie, and S. Jr, "PicoRadio Supports Ad Hoc Ultra-Low Power wireless networking," *IEEE Computer*, vol. 33, no. 7, pp. 42–48, 2000.

[18] N. D. Georganas, D. C. Petriu, and D. Makrakis, "Sensor-Based Information Appliances," *IEEE Instrumentation & Measurement Magazine*, no. December, pp. 31–35, 2000.

[19] P. Deshpande and M. S.Madankar, "Techniques Improving Throughput of Wireless Sensor Network : A Survey," *International Conference on Circuit, Power and Computing Technologies [ICCPCT]*, pp. 0–4, 2015.

[20] J. Sliva, "Technologies used in Wireless Sensor Networks," *15th International Conference on Systems,Signals and Image Processing,IEEE*, 2008.

[21] T.-w. Song and C.-s. Yang, "A Connectivity Improving Mechanism for ZigBee Wireless Sensor Networks," *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, pp. 495–500, 2008.

[22] S. C. Pedram Radmand , Alex Talevski, Stig Petersen, "Comparison of Industrial WSN Standards," *IEEE International Conference on Digital Ecosystems and Technologies*, pp. 632–637, 2010.

[23] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS : An Operating System for Sensor Networks," *In: Ambient intelligence*, vol. 2, pp. 115–148, 2005.

[24] V. T. Dunkels A, Gronvall B, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," *Proceedings of the 29th annual IEEE international conference on local computer networks. IEEE Computer Society,*, pp. 455–462, 2004.

[25] C.-c. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, "SOS - A Dynamic operating system for Sensor Networks," *Proceedings of the third international conference on mobile systems applications and services (Mobisys). ACM, New York*, pp. 163–176, 2005.

[26] S. Bhatti, J. Carlson, H. U. I. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "MANTIS OS : An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms," *Mobile Networks and Applications 10*, pp. 563–579, 2005.

[27] H. T. Kuorilehto M, Alho T, Hannikainen M, "SensorOS : A New Operating System for Time Critical WSN Applications," *Embedded computer systems: architectures, modeling, and simulation. Springer, Berlin/New York*, pp. 431–442, 2007.

[28] W. S. Networks and W. S. Networks, "Wireless Sensor Networks," in *To appear in Smart Environments:Technologies, Prorotocols and Applications*, 2004, pp. 1–18.

[29] M.-l. Messai, "Classification of Attacks in Wireless Sensor Networks," *International Congress on Telecommunication and Application*, vol. 14, no. April 2014, pp. 23–24,

2014.

[30] J. N. AL-KARAKI and A. E. KAMAL, "ROUTING TECHNIQUES IN WIRELESS SENSOR NETWORKS : A SURVEY," *IEEE Wireless Communications*, no. December, pp. 6–28, 2004.

[31] I. P. Communications, "Sensor Information Networking Architecture and Applications," *IEEE Personal Communications*, no. September 2001, pp. 52–59, 2001.

[32] T. Of, "A COMPARATIVE STUDY OF CLUSTER HEAD SELECTION ALGORITHMS IN WIRELESS SENSOR," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 2, no. 4, pp. 153–164, 2011.

[33] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Hawaii International Conference on System Sciences*, *IEEE*, 2000.

[34] K. Benki, M. Malajner, P. Planinši, and Ž. Č, "Using RSSI value for distance estimation in Wireless sensor networks based on ZigBee," in *Systems,signals and image processing,15th international conference on.IEEE*, 2008, pp. 303–306.

[35] S. Lee, J. Hong, J. Kook, and S. Lee, "T-LEACH : The method of threshold-based cluster head replacement for wireless sensor networks," *Information Systems Frontiers ·*, no. November 2009, pp. 513–521, 2017.

[36] N. Kumar, "Improved LEACH Protocol for Wireless Sensor Networks," in *Proceedings of 7 th International Conference on Wireless Communications Networking and Mobile Computing*, 2011.

[37] N. Kumar, P. Bhutani, and P. Mishra, "U-LEACH : A Novel Routing Protocol for," in *International Conference on Communication, Information & Computing Technology (ICCICT)*, 2012, pp. 1–4.

[38] S. Koteswararao, "Implementation of Multi-hop Cluster based Routing Protocol for Wireless Sensor Networks," *International Journal of Computer Applications (0975*, vol. 59, no. 8, pp. 1–5, 2012.

[39] Z. Li, "Survey on Security in Wireless Sensor Networks," *Speech English Edition of Journal of KIISC*, vol. 18, no. 6, 2008.

[40] Y. Wang, G. ATTEBURY, and B. RAMAMURTHY, "A Survey of Security Issues In Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 1–23, 2006.

[41] J. P. Walters and Z. Liang, "Wireless Sensor Network Security : A Survey," *Security in Distributed, Grid, and Pervasive Computing Yang Xiao,(Eds.)*, pp. 1–50, 2006.

[42] D. G. Padmavathi and M. D. Shanmugapriya, "A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks," *(IJCSIS) International Journal of Computer Science and Information Security*, vol. 4, no. 1, p. 9, 2009. [Online]. Available: http://arxiv.org/abs/0909.0576

[43] M. M. Patel, "Security Attacks in Wireless Sensor Networks : A Survey," *IEEE International Conference on Intelligent Systems and Signal Processing (ISSP)*, pp. 329–333,

2013.

[44] A. Ahmed, "An Evaluation of Security Protocols on Wireless Sensor Network," *TKK T-110.5190 Seminar on Internetworking*, 2009. [Online]. Available: http://www.cse.tkk.fi/en/publications/B/5/papers/ahmed{_}final1.pdf

[45] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," *ACM Journal of Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2001.

[46] P. G. Shah, "Network Security Protocols for Wireless Sensor Networks-A Survey," *In International conference on Cognitive Systems*, no. I, 2005.

[47] A. Sharma, G. Tripathi, S. Khan, and K. A. Kumar, "A Survey Paper on Security Protocols of Wireless Sensor Networks," *International Research Journal of Engineering and Technology (IRJET)*, vol. 02, no. 08, pp. 1548–1552, 2015.

[48] K. Venkatraman, J. V. Daniel, and G. Murugaboopathi, "Various Attacks in Wireless Sensor Network: Survey," *International Journal of Soft Computing and Engineering*, vol. 3, no. 1, pp. 2231–2307, 2013.

[49] T.-g. Lupu and V. Parvan, "Main Types of Attacks in Wireless Sensor Networks," *In Proceedings of the 9th WSEAS International Conference on signal,speech and image processing and 9th WSEAS Internation conference on Multimedia,internet and vidoe technologies(SSIP '09/MIV'09)*, pp. 180–185, 2009.

[50] P. Arora and A. Gupta, "A Survey on Wireless Sensor Network Security," *International Journal of Computer Science and Information Technology Research*, vol. 2, no. 2, pp. 67–76, 2014.

[51] D. Anthony and A. John, "Denial of Service in Sensor Networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[52] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, pp. 102–114, 2002.

[53] S. K. M, "Detection of Jamming Style DoS attack in Wireless Sensor Network," *IEEE International Conference on Parallel, Distributed and Grid Computing*, pp. 563–567, 2012.

[54] J. Newsome, E. Shi, D. Song, A. Perrig, J. Newsome, and A. Perrig, "The Sybil Attack in Sensor Networks : Analysis & Defenses," *Proceedings of the 3rd International symposium on Information Processing in Sensor Networks.ACM*, pp. 259–268, 2004.

[55] C. Ozturk, Y. Zhang, and W. Trappe, "Source-Location Privacy in Energy-Constrained Sensor Network Routing," *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks.ACM*, pp. 88–93, 2004.

[56] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-Aware Location Sensor Networks," *HotOS IX: The 9th Workshop on Hot Topics in Operating Systems*, pp. 163–167, 2003.

[57] S. Hussain and S. Rahman, "Using Received Signal Strength Indicator to Detect Node Replacement and Replication Attacks in Wireless Sensor Networks," *Data Mining*,

Intrusion Detection, Information Security and Assurance, and Data Networks Security, vol. 7344, pp. 1–11, 2009.

[58] G. Dass and R. Singh, "Detection and Avoidance of Clone Attack in WSN Using Neighbor Witness Node," *International Journal of Science and Research (IJSR)*, vol. 5, no. 1, pp. 424–428, 2016.

[59] B. Parno and A. Perrig, "Distributed Detection of Node Replication Attacks in Sensor Networks âĹŮ," *Proceedings of the IEEE Symposiumon Security and Privacy (IEEE S and P'05)*, pp. 267–281, 2005.

[60] M. Gupta and M. Lumb, "IDS for Node Replicas at Distributed & Centralized Levels in Mobile WSN," *International Journal of Engineering Research & Technology*, vol. 3, no. 8, pp. 494–498, 2014.

[61] R. Brooks, S. Member, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the Detection of Clones in Sensor Networks Using Random Key Predistribution," *IEEE Transactions on Systems, Man, and Cybernetics Part C:Systems and Humans*, vol. 37, no. 6, pp. 1246–1258, 2007.

[62] K. X. K. Xing, F. L. F. Liu, X. C. X. Cheng, and D. Du, "Real-Time Detection of Clone Attacks in Wireless Sensor Networks," *2008 The 28th International Conference on Distributed Computing Systems*, pp. 3–10, 2008.

[63] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "Distributed Detection of Clone Attacks in Wireless Sensor Networks," *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, vol. 8, no. 5, pp. 685–698, 2011.

[64] W. Znaidi, M. Minier, and S. Ubeda, "Hierarchical node replication attacks detection in wireless sensors networks," *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 82–86, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5450196

[65] D. Xia, "NEAR-OPTIMAL NODE CLUSTERING IN WIRELESS SENSOR NETWORKS FOR ENVIRONMENT MONITORING Natalija Vlajic," *International Conference on Advanced Networking and Applications(AINA'07)*, no. 1, pp. 632–641, 2007.

[66] A. Kumar, M. Ashok, and K. Turuk, "A Zone-Based Node Replica Detection Scheme for Wireless Sensor Networks," *Wireless Personal Communications*, pp. 601–621, 2013.

[67] C. M. Yu, C. S. Lu, and S. Y. Kuo, "CSI: Compressed sensing-based clone identification in sensor networks," *2012 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2012*, no. March, pp. 290–295, 2012.

[68] G. Mao and B. D. O. Anderson, "Wireless sensor network localization techniques," *ELSEVIER's Computer Networks 51*, pp. 2529–2553, 2007.

[69] Y. C. Hu and Æ. I. Stojmenovic, "Hierarchical geographic multicast routing for wireless sensor networks," *Wireless Netwoek*, vol. 16, pp. 449–466, 2010.

[70] Y. Wu, Z. Chen, Q. Jing, and Y.-c. Wang, "LENO : LEast Rotation Near-Optimal Cluster Head Rotation Strategy in Wireless Sensor Networks," *IEEE 21st International Conference on Advanced Networking and Applications(AINA'07)*, 2007.

[71] "OMNeT++ (Objective Modular Network Test-bed in C++)," p. https://www.omnetpp.org, 2017.

Appendix

9.1. Simulation Coding

9.1.1. Definition(NED) of Modules

==========

- **Sensornode NED definition**

```
simple SensorNode
{
    parameters:
        double xpos = uniform(1, 1000);
        double ypos = uniform(1, 1000);
        int hcount=default(0);
        int ID;
        int gateID;
        double energy;
        double simstart;
        volatile int pkLenBits @unit(b);  // packet length in bits
        @signal[state](type="int");
        @signal[transmit](type="long");
        double range = default(200); // Range in which sensors will receive over-threshold signal (simulation)
        double radioDelay @unit(s);
        double txRate @unit(bps);         // transmission rate
       // string signal = default("bit string representing signal");
        @class(SensorNode);
        int angle;
        double speed @unit("mps") = default(10mps);
        int acceleration;
        double timeStep @unit("s") = default(0.1s);
        @signal[packetSent](type="long");
        @signal[packetReceived](type="long");
        @signal[deadSensorNodes](type="long");
        @signal[roundsDeadDone](type="long");

        @statistic[packetSent](title="Packet Sent";source="packetSent";record=vector?,count?,mean?; interpolationmode=none);
        @statistic[packetReceived](title="Packet Received";source="packetReceived";record=vector?,count?,mean?; interpolationmod
        @statistic[deadSensorNodes](title="Number Dead Nodes";source="deadSensorNodes";record=vector?,count?,mean?; interpolatio
        @statistic[roundsDeadDone](title="Round Of Dead";source="roundsDeadDone";record=vector?,count?,mean?; interpolationmode=
    gates:
        inout port[] ;
        input in[] ;
        output out[];
        inout sports @directIn ;
}
```

- **Network NED definition**

==========

```
package ned.ned;import ned.IUnidirectionalChannel;
module WSN extends wsn_attack
{
    parameters:
        int numNodes;
        int numReplica;
        int totalNodes;
        double trRange;
        int rounds;
        int frames;
        double sunDuration;
        int sunNodes;
        double xMax;
        double yMax;
        int clones = default(0);
        volatile bool cloneFound = default(false);
        volatile bool foundClone = default(false);
        double txRate @unit(bps);          // transmission rate
        @labels(SensorNode);
        @figure(textFormat = "packet received");
        int updateinterval;
        double nodeSeparation @unit("m")=default (40m);
         @display("bgb=2000,1000;bgg=50,2,grey60;i=block/network,bgi=,s");
    submodules:
        bs: BaseStation {
            parameters:
        }
        Sensor[totalNodes]: SensorNode {
            parameters:
                @display("i=abstract/person;p=0,0");
        }
        kmanagement: KeyManagement {
            @display("p=1400,900;i=block/encrypt_l,red;is=l");
        }
        cloneDetection: Detection {
            @display("p=1800,500;i=block/rx_l");
        }
    connections allowunconnected:
```