

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

**Wormhole Attacks Detection in Wireless Sensor Networks by
Analyzing Transmission range and Residual Energy of Nodes**

By: Kemal Hussen

Advisor: Dr. Fisseha Bayu (PhD.)

Co-advisor: Mr. Nahil Kebede (MSc.)

**A THESIS SUBMITTED TO SCHOOL OF GRADUATE STUDIES, JIMMA UNIVERSITY,
JIMMA INSTITUTE OF TECHNOLOGY, FACULTY OF COMPUTING AND
INFORMATICS IN PARTIAL FULFILLMENT OF THE DEGREE OF MASTER OF
SCIENCE IN COMPUTER NETWORKING**

4 December 2022

Jimma, Ethiopia

Approval sheet

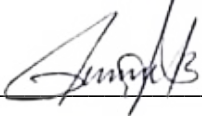
This Independent Research entitled “*Wormhole attacks Detection in Wireless Sensor Network by analyzing Transmission range and Residual energy of nodes*” has been read and approved as meeting the preliminary research requirements of the School of Computing in partial fulfillment for the award of the degree of Master in Computer Networking, Jimma University, Jimma, Ethiopia.

Principal Investigator: Mr. Kemal Hussen

Sign _____ 

Dr. Fisseha Bayu (Ph.D.)
Principal Advisor

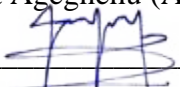
Mr. Nahil Kebede (MSc.)
Co-Advisor

Sign _____ 

Sign _____ 

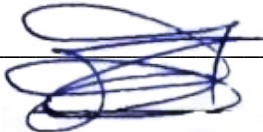
External Examiner

Dr. Mekuanint Agegnehu (Assoc. Professor)

Sign: _____ 

Internal Examiner

Mr. Kebebew Ababu (Ass. Professor)

Sign: _____ 

Chair Person

Mr. Alehegn Minale(MSc.).

Sign: _____

Acknowledgment

Firstly, I would like to thank God for everything. I would also like to show my sincere gratitude to my advisor Dr. Fisseha Bayu for his initial guidance and continuous support during my thesis progress. I would like to thank Mr. Nahil Kebede for his constant help and he is always available to discuss and provide insightful comments on my work throughout my program. It was a pleasure working with him and also a wonderful learning experience.

List of Figures

| | |
|--|----|
| Fig 1. 1 Wormhole attack in WSN | 2 |
| Fig 2. 1 Functional components of sensor nodes | 7 |
| Fig 2. 2 Sensing and communication range | 8 |
| Fig 2. 3 Classification of wormhole attacks | 13 |
| Fig 4. 1 location of nodes in 2D graph | 22 |
| Fig 4. 2 distance between nodes on fixed transmissionrange | 23 |
| Fig 4. 3 Architecture of wormhole attacks detection in WSN | 24 |
| Fig 4. 4 Source initialization | 27 |
| Fig 4. 5 Wormhole detection and prevention flowchart | 28 |
| Fig 5. 1 AODV without wormhole simulation | 40 |
| Fig 5. 2 AODV with wormhole simulation | 41 |
| Fig 5. 3 AODV with WDPT simulation | 42 |
| Fig 5. 4 Throughput Received by node 4 from node 0 | 43 |
| Fig 5. 5 Throughput Received by node 1 from node 0 | 43 |
| Fig 5. 6 Throughput received by node1 and node4 | 44 |
| Fig 5. 7 Detection rate comparison at different hops | 47 |

List of Tables

| | |
|---|----|
| Table 2. 1 Placement of sensors and its coverage area | 8 |
| Table 5. 1 simulation parameters | 39 |
| Table 5. 2 route table format | 40 |
| Table 5. 3 route table without Wormhole attack | 40 |
| Table 5. 4 route table with wormhole | 41 |
| Table 5. 5 WDPT route table | 42 |
| Table 5. 6 Average Residual Energy | 45 |
| Table 5. 7 Throughput and PDR | 46 |
| Table 5. 8 Effect of wormhole attack in AODV | 46 |

Acronyms, Abbreviation, and Terminology

| | |
|--------|--|
| WDPT: | Wormhole Detection by analyzing Transmission range |
| WSN: | Wireless Sensor Network |
| AODV: | Ad-hoc On-Demand Distance Vector |
| DoS: | Denial of Service |
| DSR: | Dynamic Source Routing |
| MANET: | Mobile Ad-hoc Network |
| MAC: | Medium Access Control |
| NS2: | Network Simulator version-2.0 |
| CIA: | Confidentiality Integrity and availability |
| HMTI: | Hello Message Time Interval |
| RTT: | round-trip travel time |
| NAM: | Network Animator |
| DR: | Detection Rate |
| SN: | Sensor Node |
| PDR: | packet delivery ratio |

Table of Contents

| | |
|--|----|
| CHAPTER ONE | 1 |
| INTRODUCTION | 1 |
| 1.1 Background | 1 |
| 1.2 Statements of Problem | 2 |
| 1.3 Objectives..... | 4 |
| 1.3.1. general objective..... | 4 |
| 1.3.2. Specific Objectives | 4 |
| 1.4. Methods..... | 4 |
| 1.5. Scope and Limitations of the study | 5 |
| 1.6. Applications of the Result..... | 5 |
| 1.7 Thesis Organization..... | 6 |
| CHAPTER TWO | 7 |
| LITERATURE REVIEW | 7 |
| 2.1 Overview of Wireless Sensor Network..... | 7 |
| 2.2 Sensor node Deployment in WSN | 8 |
| 2.3 Communication protocols for wireless sensor networks | 9 |
| 2.4 Sensor Network Security issue..... | 10 |
| 2.5 categories of attacks in Wireless Sensor Network | 10 |
| 2.5.1 Denial of Service | 10 |
| 2.5.2 Sybil Attacks..... | 11 |
| 2.5.3 Blackhole Attacks..... | 12 |
| 2.5.4 Wormhole attacks | 12 |
| 2.6 Wormhole Attack Classification | 12 |
| 2.7 Detection and Avoidance of Wormhole Attacks..... | 13 |

| | |
|--|----|
| CHAPTER THREE | 15 |
| RELATED WORKS | 15 |
| 3.1 Packet Leashes and Position of Nodes | 16 |
| 3.3. Connectivity and Neighborhood-Based Approaches | 18 |
| 3.4. Graphical and Topological Information-Based Approaches..... | 19 |
| 3.5. Routing Algorithm-Specific Approaches..... | 19 |
| 3.6. Special Hardware-Based Approaches | 19 |
| 3.7. Hop-Count Analysis Technique | 20 |
| CHAPTER 4 | 21 |
| DESIGN OF THE PROPOSED SOLUTION..... | 21 |
| 4.1 Architecture of Detection and prevention of Wormhole Attacks | 24 |
| 4.2 Out-Of-Band Wormhole Detection..... | 29 |
| CHAPTER FIVE | 31 |
| IMPLEMENTATION AND RESULT EVALUATION | 31 |
| 5.1 Overview | 31 |
| 5.2 Simulation Tools and Development Languages..... | 32 |
| 5.2.1 OMNET++ | 32 |
| 5.2.2 NS-3 (Network Simulator-3)..... | 33 |
| 5.2.3 GLOMOSIM | 33 |
| 5.2.4 NS-2(Network Simulator-2)..... | 34 |
| 5.2.5 Trace data analyzing applications in NS-2..... | 34 |
| 5. 3 Network Components in a Sensor Node | 35 |
| 5.4. Structure of Trace Files | 38 |
| 5.5. Implementation Details | 39 |
| 5.5.1 Wormhole Detection Rate | 47 |

| | |
|----------------------------------|----|
| CHAPTER 6 | 49 |
| CONCLUSION AND FUTURE WORK | 49 |
| 6.1 Conclusion | 49 |
| 6.2 Contributions..... | 49 |
| 6.3 Future Works | 50 |
| REFERENCES | 51 |
| Appendix..... | 56 |

Abstract

*Wireless Sensor Networks have made significant progress and have emerged as an important study topic in wireless and distributed networks. Wireless sensor networks (WSN) are made up of a large number of tiny and inexpensive devices called sensor nodes. The sensor nodes are capable of detecting, actuating, and regulating the information gathered. There are several research challenges and issues with WSN such as security, power efficiency, scalability, responsiveness, and reliability. security becomes a key prerequisite for modern-age applications. Weak security or absence of security may not only conciliate classified information but also makes them accessible for malicious attacks. The network layer is vulnerable to different types of attacks like a Sinkhole, Wormhole, Sybil, Selective Forwarding, Hello Flood, Black Hole, greyhole, and so on. This paper deals with the detection and prevention of an attack on the network layer called a **wormhole attack**. A wormhole attack is one of the most popular and serious attacks in the wireless sensor network. It is a particularly damaging attack on routing protocols for specially designated systems in which two or more collaborating attackers record packets at one location and tunnel them to another for replay at that remote location. In This paper, we make a literature review of the detection and prevention of wormhole attacks. Also proposed transmission range-based and residual energy mechanisms for the detection of wormhole attacks. When the source node received RREP, it tracks the location of nodes on the route using GPS and records the actual distance between them and the minimum number of hops. Simulation results are used with the NS-2 simulator and our method has been evaluated in terms of detection rate, packet delivery ratio, throughput, and residual energy compare to a network without or with an attack. And results show that the detection rate of our method is 89.5% of the total adversary attacks conducted.*

Keywords - Security, Wormhole, Tunnel, transmission range, WSN, NS-2,GPS

CHAPTER ONE

INTRODUCTION

1.1 Background

Wireless sensor networks [9] are made up of plenty of small, low-power devices that integrate limited computation, sensing, and radio communication capabilities. They provide adaptable infrastructures for a wide range of applications, including healthcare, industrial automation, surveillance, and military applications. These networks' security issues occur due to a lack of a trusted centralized authority, easy packet loss due to shared wireless medium, unpredictable topology, poor bandwidth, and battery capacity. A wormhole attack is a potential threat in ad hoc networks [1]. During the attack [3] an attacker node captures data from one point in the network and tunnels them to a faraway attacker node, which repeats them locally., this is illustrated in Figure 1.1

In wormhole attacks, an attacker captures a packet, or specific bits from a packet, at one point in the network, tunnels it (perhaps selectively) to another point in the network, and replays it there. These attacks can cause substantial danger to wireless networks, particularly to various ad hoc network routing algorithms and location-based wireless security solutions. The wormhole puts the attacker in an extremely strong position, allowing the attacker to obtain unauthorized access, disrupt routing, or conduct a Denial-of-Service (DoS) attack.

Attackers can launch wormhole attacks in WSNs without exposing their identity. Most routing protocols are vulnerable to this attack, including AODV (Ad hoc On-demand Distance Vector) and DSR (Dynamic Source Routing). Because the attackers in a wormhole assault are directly connected through a tunnel, they may communicate at a faster rate than other nodes in the sensor network

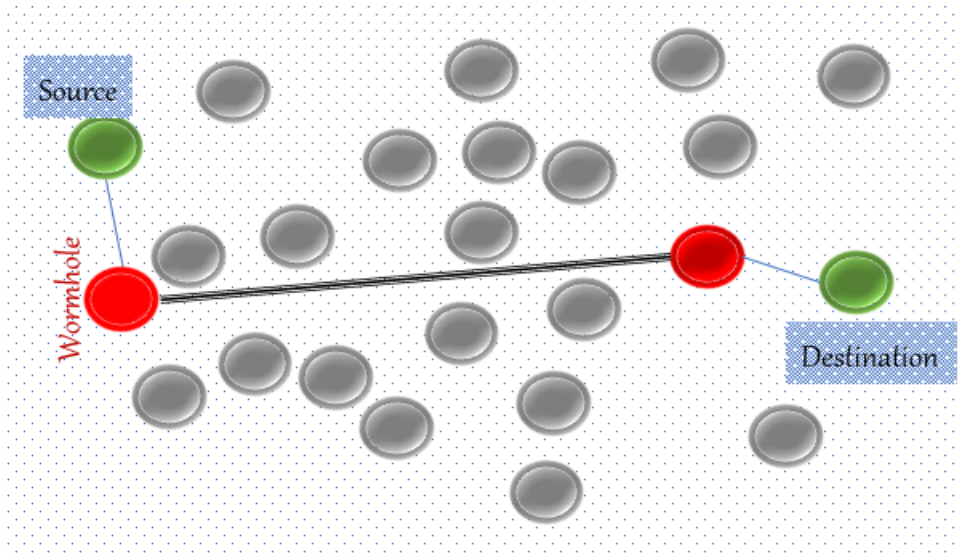


Fig 1. 1 Wormhole attack in WSN

In this paper, we tried to enhance a *wormhole* attack detection mechanism for WSNs by analyzing the transmission range and residual energy of nodes that are connected to the network.

1.2 Statements of Problem

There are several research challenges and issues with WSNs such as security, power efficiency, scalability, responsiveness, and reliability [4]. security becomes a key prerequisite for modern-age applications [29]. Weak security or absence of security may not only conciliate classified information but also makes them accessible for malicious attacks. In the WSN, several anomalies can occur due to their lack of processing and communicating capability, limited storage capacity, transmission range, bandwidth, and energy. These networks are usually deployed in a remote area and left unattended; they should be equipped with security, and defenses against attacks such as capturing nodes, physical tampering, eavesdropping, denial of service, etc. Unfortunately, traditional high-overhead security mechanisms are not feasible for resource-constrained sensor nodes.

One of the major issues with wireless sensor networks is upholding confidentiality. A wireless sensor network should not leak out any of its credentials even when sensors are read by their neighbor nodes. They use encryption algorithms for privacy conservation. Encryption mechanisms

are very awkward as they generate security overhead and enlarge packet size for that reason. They also increase energy utilization due to encryption and decryption procedures and network traffic.

Wormhole attacks are among the most dangerous threats in WSN [8]. In general, two or more attacker nodes will establish a secret path known as a *tunnel*. In a wormhole attack, an attacker obtains packets at one point in the network and sends them through their tunnel to another point in the network and reply each other through an established wormhole channel called a *tunnel*. For tunneled distances greater than a single hop's usual wireless transmission range, the attacker may easily have the tunneled packet arrive with a better metric than a conventional multi-hop route by using a single long-range directional wireless link.

To detect wormhole attacks, many researchers have been developing so many detection techniques [8], adding the location of nodes and packet sent time with routing packets (*Packet leashes*) [14], *Delphi* (Delay-per-hop) [6], *Statistical Analysis method* (hypothesis testing) [18] and so on.

In the *Statistical Analysis method*, wormhole nodes assumed that they can't modify and drop packets [48]. But one of the most critical problems that make Wormhole attack detection is harder, attacker nodes can *modify* or *drop* routing packets or data. When we come to the *Delphi* method, it checks the delays on each link when RREP is received, but some wormhole attacks can transfer data at a faster rate than normal nodes through its *tunnel* [14]. On the other hand, delay can be occurred due to low transmission energy to transfer data from one node to another, routing overheads, and congestion. Due to this detection rate of the *Delphi* method is prone to false positive results. *Packet leashes* (Hash-based Compression Function (HCF)) method implement *HCF* for each sending of data. When data is intended to send n times, the HCF function also will be implementing n times. Since sensor nodes are powered by batteries, the lifetime of sensor nodes will decrease because of the increasing energy consumption to implement HCF.

To overcome these challenges, in recent years, there have been various attempts to propose enhanced wormhole attack detection systems in WSN.

Our work focused on developing wormhole attack detection with a better detection rate, and less analyzing time. We will analyze the transmission range of each node used to forward packets from source to destination and we will also analyze the residual energy of nodes used for transmitting

the generated packets from source to destination as well as we will count the number of routing hops from source to destination.

Generally, the following research questions are to be answered in this thesis: **To check the presence of wormhole nodes in WSN and to prevent wormhole attacks in WSN if it exists.**

1.3 Objectives

1.3.1. general objective

The general objective of this thesis is wormhole attack detection in wireless sensor networks by analyzing the transmission range and residual energy of nodes.

1.3.2. Specific Objectives

To accomplish our general objective, we have the following specific objectives:

- ⇒ Investigate and recognize the current wormhole attacks detections schemes.
- ⇒ Design architecture for our new proposed wormhole attack detection scheme.
- ⇒ Implement the proposed solution in the wireless sensor network scenario on the simulator NS-2 environment.
- ⇒ Test and evaluate through simulation the detection rate of the proposed solution to demonstrate that it enhances the existing system wormhole attack detection.
- ⇒ Compare and contrast the new scheme with existing schemes.

1.4. Methods

Literature Review: - several studies and explorations will be made on the areas related to wormhole attack detection and prevention on WSN. This will be accomplished by reading different books, journals, or conference papers that have been done so far with different approaches, to have a sufficient understanding of the problem. Techniques and approaches appropriate for the development of a routing algorithm for AODV routing protocol and other routing protocols in WSNs will also be reviewed as well. After a deep understanding of the problem, we will propose a better wormhole detection technique in WSN.

Design and Implementation: - While we do this study, we will use different algorithms to achieve the specified objectives and we use the network simulation toolkit as a working environment. This study involves the development of an enhanced wormhole detection by analyzing the transmission

range and the residual energy of each node to transmit packets for the current Wormhole attack problem in the AODV routing protocol by adding an enhanced improvement technique.

Software simulation: - we will use an NS2 network simulator for simulation, analysis, and comparison of wormhole attacks in the WSN AODV routing protocol. We have tried to discuss the theoretical background of each wormhole attack in WSN and how well it represents real life by deploying wormhole nodes in the MAC layer and we use AODV routing protocol for this scenario, which resembles real-life cases.

1.5. Scope and Limitations of the study

- ❖ The proposed wormhole attack detection methods rely on distances between nodes, residual energy of nodes that participated in route requests, and route reply in the network for static WSN.
- ❖ The method we proposed doesn't address
 - ⇒ More than one wormhole link
 - ⇒ WSN with mobility
- ❖ We designed and implemented a wormhole detection technique that identifies wormhole nodes in WSN and we also analyzed with and without wormhole during AODV routing using RREQ and RREP and by analyzing the energy used by each node. There are two classification networks. The one that has a WSN has a wormhole node and a WSN that hasn't a wormhole node.

1.6. Applications of the Result

Detecting wormhole attacks will significantly contribute to the area of wireless communication in WSN for the removal of packet loss and route disruptions during communication because as long as the wormhole detection method is enhanced an organization or a user will be satisfied with the service. This work will facilitate the services of WSN applications and the most important application areas that will benefit from this work are emergency scenarios like military environments, earthquake monitoring, health monitoring, weather forecasting, and maybe in education.

According to Buratti et al. [38], the various conceivable applications of WSNs to every sector globally are essentially boundless, from environmental monitoring and management to medical

and healthcare services, as well as other aspects such as positioning and tracking, localization, and logistic. Strappingly, it is imperative to emphasize that the benefits and applications affect the choice of wireless machinery to be employed.

As soon as the requirements of the application are set, the network designers need to select and choose the machinery which allows the gratification of these requirements. Hence, the knowledge of the structures, benefits, and difficulties of the various pieces of machinery is fundamental. As a result of the significance of the relationship between the requirements for application and the machinery, this section will attempt to briefly give an outline of some of the utmost applications of WSNs

1.7 Thesis Organization

The rest of this paper is organized as follows. The second chapter discusses a literature review on wormhole attacks, the third chapter reviews the related work on wormhole attack detection and prevention. In the fourth chapter, we describe the proposed wormhole attack detection method and essential assumptions. Chapter five explains the simulation environment that we used and the result of our method. Finally, in chapter six, we conclude this paper and outline our future work

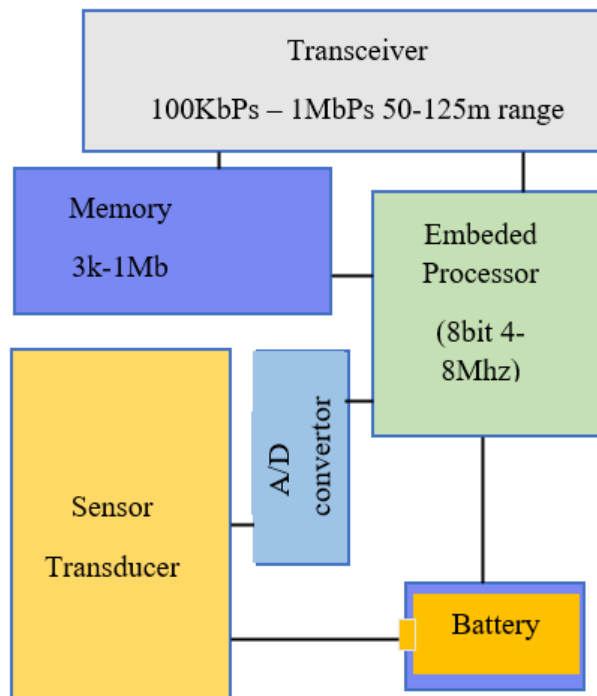
CHAPTER TWO

LITERATURE REVIEW

In this chapter, we discuss different wormhole attacks and detection in wireless sensor networks on Ad-hoc On-Demand Distance Vector (AODV) routing protocol, provide a comparison between them and finally enumerate research work that has been carried out in the design, implementation, and testing wormhole attack detection in WSN network and prevention.

2.1 Overview of Wireless Sensor Network

A Wireless sensor network is a multi-hop wireless network that is established by a group of sensor nodes on a shared wireless channel [2]. The nodes communicate with each other and exchange network information as needed, and network topology changes could occur randomly, rapidly, frequently, and unpredictably. As a host, a node functions as a source and a destination in the network and as a router; nodes act as intermediate bridges between the source and the destination giving store and forward services to all the neighboring nodes throughout the communication.



As we can see from fig 2.1, sensor node has six main components to sense, actuate and transmit data in real world.

For modern sensor sensing, receiving and transmission range may above 125m.

Fig 2. 1 Functional components of sensor nodes

Each node has a limited range of detection. The maximum distance a node can detect is called the sensor's sensing radius, while the area within the sensing radius is called the coverage area, see Fig 2.2

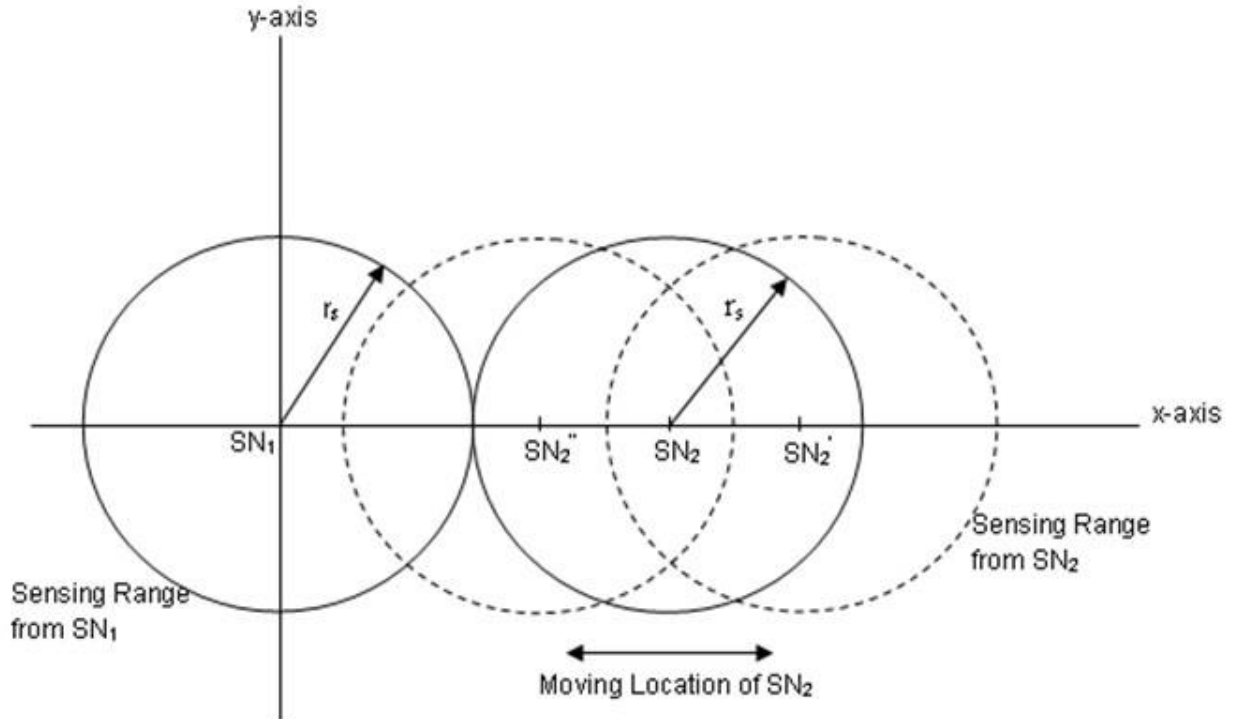


Fig 2. 2 Sensing and communication range

2.2 Sensor node Deployment in WSN

To create sensor network nodes must be able to communicate with each other for sending and receiving data gathered by those nodes. Table 2.1 shows the sensor node placement and its coverage area.

| Placement | Distance between adjacent sensors | Sensing Area to be covered by each sensor | Total Sensing Area covered by N-Sensors |
|-------------|-----------------------------------|---|---|
| Rectangular | r | r^2 | $N \cdot r^2$ |
| Triangular | r | $\frac{\sqrt{3}}{4} r^2$ | $N \cdot \frac{\sqrt{3}}{4} r^2$ |
| Hexagon | r | $\frac{3\sqrt{3}}{4} r^2$ | $N \cdot \frac{3\sqrt{3}}{4} r^2$ |

Table 2. 1 Placement of sensors and their coverage area

2.3 Communication protocols for wireless sensor networks

Studies conducted on resource management in wireless sensor networks have identified energy-efficient routing protocols as one of the energy-saving mechanisms that can be used to manage the consumption of networks' available energy and extend network lifetime.

Routing protocols assist in finding paths for the transmission of sensed events, and they must be able to extend the lifetime of a network despite some of the limitations of sensor nodes in a network and the harsh environments in which the sensor nodes are to operate.

A wireless communication network is formed in an ad hoc manner where sensor nodes can organize themselves with no proper coordination, this is found in most WSNs applications. The source of power for the sensor nodes is a battery, which is usually not rechargeable or replaceable especially when the sensor nodes are expected to operate with no human intervention for a longer period during the application [44,45]. Careful resource management is a prime concern in the design of wireless sensor networks. It can be achieved through energy-saving techniques such as Radio Optimization, Data Reduction, Sleep or Wake-up methods, Energy Efficient routing protocols, and Energy Harvesting [46].

- **AODV protocol**

Large-scale sensor networks are susceptible to link failures due to the long transmission range and deployment of many sensor nodes. With this in mind together with the ad-hoc nature of the deployment of sensor nodes in sensor networks, AODV [47] is a suitable communication protocol for these networks. AODV allows the sensor network to adapt to dynamic link states quickly. Sensor nodes can respond timely to frequent changes in network topology and breakages in link connectivity. This is made possible using destination sequence numbers that always ensure free loops in the network. Routes in AODV are discovered only when they are required. Periodic HELLO messages are utilized in the original AODV, to assess if links to neighboring nodes are valid. In AODV, RREP packets are not generated by forwarding nodes even if they have valid routes and hence avoid adding multiple replies overheads. Cross-layer techniques that help to avoid high packet loss paths are also included in this version.

2.4 Sensor Network Security issue

Authentication: - used to verify the identity of the message sender at receiver.

Confidentiality - it ensures that the content of the data is accessed only by authorized nodes.

Integrity - it guarantees that should a message have its content modified during the transmission; the receiver can identify these alterations.

2.5 categories of attacks in Wireless Sensor Networks

2.5.1 Denial of Service

Denial of Service (DoS) is any action that makes network service stop or eliminates a network's capacity to perform its expected function. A denial-of-service (DoS) attack is a type of cyber-attack in which a malicious actor aims to render a computer or other device unavailable to its intended users by interrupting the device's normal functioning. DoS attacks typically function by overwhelming or flooding a targeted machine with requests until normal traffic is unable to be processed, resulting in denial-of-service to additional users. A DoS attack is characterized by using a single computer to launch the attack.

The primary focus of a DoS attack is to oversaturate the capacity of a targeted machine, resulting in denial-of-service to additional requests. The multiple attack vectors of DoS attacks can be grouped by their similarities.

DoS attacks typically fall into 2 categories:

❖ **Buffer overflow attacks**

An attack type in which a memory *buffer overflow* can cause a machine to consume all available hard disk space, memory, or CPU time. This form of exploit often results in sluggish behavior, system crashes, or other deleterious server behaviors, resulting in a denial of service.

❖ **Flood attacks**

By saturating a targeted server with an overwhelming number of packets, a malicious actor can oversaturate server capacity, resulting in a denial of service. For most DoS flood attacks to be successful, the malicious actor must have more available bandwidth than the target.

2.5.2 Sybil Attacks

The Sybil attack is a massive destructive attack against the sensor network where numerous genuine identities with forged identities are used for getting illegal entry into a network.

A Sybil attack uses a single node to operate many active fake identities (or Sybil identities) simultaneously, within a peer-to-peer network. This type of attack aims to undermine the authority or power in a reputable system by gaining the majority of influence in the network. The fake identities serve to provide this influence.

A successful Sybil attack provides threat actors with the ability to perform unauthorized actions in the system. For example, it enables a single entity, such as a computer, to create and operate several identities, such as user accounts and IP address-based accounts. All of these fake identities, trick systems, and users into perceiving them as real. The name of this attack was inspired by a 1973 book called *Sybil*; a woman diagnosed with dissociative identity disorder. In the context of attacks, the term was originally coined by Brian Zill, and initially discussed in a paper by John R. Douceur, both at Microsoft Research.

Here are several problems a Sybil attack may cause:

- ✚ **Block users from the network**—a Sybil attack that creates enough identities enables threat actors to out-vote honest nodes and refuse to transmit or receive blocks.
- ✚ **Carry out a 51% attack**—a Sybil attack that enables one threat actor to control over half (51% or more) of a network's total hash rate or computing power. This attack damages the integrity of a blockchain system and can potentially cause network disruption. A 51% attack can modify the order of transactions, reverse the actor's transactions to enable double-spending, and prevent the confirmation of transactions.

The main goal of a Sybil attack on a blockchain network is to gain disproportionate influence over decisions made in the network. The attacker creates and controls several aliases to achieve this effect.

2.5.3 Blackhole Attacks

A black hole attack occurs, when an intermediary captures and re-programs a set of nodes in the network to block/drop the packets and generates false messages instead of forwarding correct/true information toward the base station in the wireless sensor network.

The black hole attack is one of the well-known security threats in wireless mobile ad hoc networks. The intruders utilize the loophole to carry out their malicious behaviors because the route discovery process is necessary and inevitable. A blackhole attack increases the number of drops packets and decreases the packet delivery ratio on MANET performance. After applying multiple numbers of black hole nodes on the network, drop packets will be increased and the packet delivery ratio drops off.

2.5.4 Wormhole attacks

A wormhole attack is described as a malicious behavior that undermines network security or misroutes or provides misleading route information to legitimate nodes. It results in the loss of a network or computer system's CIA. For example, an intrusion may compromise a network's CIA by obtaining root-level access and then altering and stealing network information. In today's world of ever-increasing Internet connection, there is a continuing threat of intrusion, denial of service assaults, or other abuses of computer and network resources [8]. To address these issues, network security solutions such as firewalls, encryption, antivirus, and so on.

Intrusion detection systems (especially wormhole attack detection) monitor the events that occur in a computer system or network to analyze the patterns of wormhole attacks [2].

2.6 Wormhole Attack Classification

classification of wormhole attacks makes it easier to design measures of protection and detection. We classified wormholes into three kinds according to whether the attackers were visible on the route: closed, partly open, and open [2]. The three forms of wormhole attacks are depicted in Figure 2.3.

Open Wormhole attack: In this kind of wormhole, the intruders include themselves in the RREQ packet header after performing route discovery. Other's node is aware that the attacker node is on the route, but they mistake them for immediate neighbors.

Closed Wormhole Attack: The intruders do not alter the packet's content, even if it is a route discovery packet. Instead, they simply forward the packet from one side of the wormhole to the other and rebroadcast it.

Half-open wormhole attack: Following the route discovery method, one side of the wormhole does not alter the packet and only the other side modifies the packet.

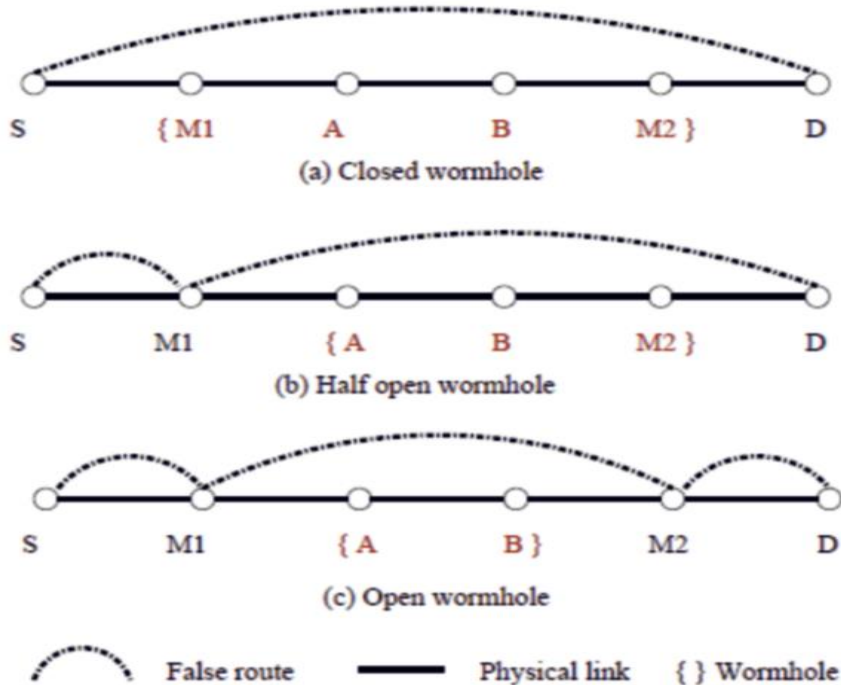


Fig 2. 3 Classification of wormhole attacks

2.7 Detection and Avoidance of Wormhole Attacks

For the past few years, the main area of research is the detection of wormhole attacks. The most important task is to discover the occurrence of wormholes in the system [10]. Detection of a wormhole based on the Hello control messages [22]. With the use of OLSR specifications, the percentage of HELLO Message Timing Intervals (HMTIs) lies in a range enclosed by the amount of jitter. A range $R = [T - \delta, T + \delta]$ has been defined. If HMTI is in the range R , then it will be considered to be valid; otherwise, it is said to be out of protocol. A second check is made whenever the HMTI packet behavior is doubtful. On the other side, a badly performing node would get

coupled with it a comparatively large number of repeat packets, which would not be the case with an attacking node. In this way, the false positive alarms problem gets negotiated.

A new protocol known as Multi-path Hop-count Analysis (MHA) is proposed based on hop-count analysis to stay away from wormhole attacks [6]. It is supposed that a very low or very high hop count is not good for the network. The uniqueness of the hop-count analysis for detecting wormhole nodes is yet uncertain.

Wormhole nodes are detected by assuming that wormhole attacks have longer packet latency as compared to the normal wireless propagation latency in a single hop [8]. As the route during the wormhole seems to be shorter, various new multi-hop routes are also channeled in the direction of the wormhole which leads to longer queuing delays. The links having delays are considered to be doubtful links, as the delay might also take place due to congestion as well as intra-nodal processing. The OLSR protocol is used for routing. This approach aims to sense the suspicious link and authenticate them in a two-step process that is described below.

in the first step, Hello packets have been sent to all the nodes that are within their transmission range. As soon as the receiver receives the Hello message, then it records the address of the sender and the time delay Δ left until it will be programmed to send its next Hello message. The node attaches the address of the sender and their respective values of time delay Δ that has been recorded for a piggybacked reply. When a Hello reply is received by a node, then it checks for the information related to any of its outstanding requests. But if no such information is there, then it will suppose it is like any other control packet. Otherwise, the node checks the arrival time of the Hello reply message to notice whether it is arrived within its scheduled timeout interval by considering the time delay Δ that occurs at the receiver side. If the arrival time is within its timeout interval, then the link between itself and the node is taken to be safe, otherwise doubtful and communication to that node is terminated by the sender until the verification process gets over.

CHAPTER THREE

RELATED WORKS

Wormhole attack detection in WSNs has been investigated by researchers and several algorithms have been proposed. In this chapter, we review and discuss published papers that are particularly related to wormhole attack detection in WSNs in AODV routing protocols.

Several researchers have proposed various solutions to combat wormhole attacks in WSNs over the last few years. This section discusses various wormhole detection techniques and their features and classifications. Hu et al. [5] proposed a method for detecting wormhole attacks using a temporal or geographical leash. Hu and Evans [6] demonstrated a method for avoiding wormhole attacks using directional antennas that sense the direction from which data is received. Khalil and Shroff [7] proposed LITEWOP, a method for detecting wormhole attacks that involves keeping and sharing encrypted lists of neighboring nodes. Muhammad Imran et al. / *Procedia Computer Science* 56 (2015) 384 – 390 Chiu and Lui [8] proposed a technique called Delphi (Delay Per-Hop Indication) to prevent wormhole attacks when using the AODV routing protocol. Using RREQ, RREP, and time duration, the technique attempts to avoid wormholes. Yun et al. [7] proposed a technique called WODEM (Wormhole Defense Mechanism) that uses a detector node with GPS technology and the ability to transmit data at various power levels. Choi et al. [10] proposed a WAP (Wormhole Attack Prevention) algorithm to prevent wormhole attacks in WSNs. Nodes in the network keep a neighbor table that records the sending and receiving times of RREQ as well as the node's suspected value. Hayajneh et al. [9] proposed the DeWorm protocol, which finds alternate routes to the destination that avoid wormhole nodes. Azer et al. [10] created a prevention and detection technique based on the principles of a social science theory known as diffusion of innovations. Alam and Chan [13] created RTT-TC, a wormhole detection technique based on round-trip time and topological comparisons. WARP is a technique presented by Su [14] to avoid wormhole attacks (Wormhole Avoidance Routing Protocol). Shi et al. [13] proposed a novel technique for detecting wormhole attacks in wireless sensor networks. The technique is divided into three phases: location, detection, and bidirectional location. If the previous stage produces a suspicious node, each of these phases is initiated. Gupta et al. [14] proposed WHOP, a wormhole detection protocol that is a modification of the AODV protocol. Shin and Halim [2] proposed a method

for detecting and isolating wormhole nodes based on route redundancy and time-based hop calculation. Khan and Islam [15] proposed a method for self-sufficient wormhole attacks based on the DSDV protocol that detects suspicious links by modifying the routing table. Chourasia and Singh [16] proposed the modified wormhole detection AODV protocol, which detects wormhole attacks by utilizing the number of hops and delay of each node in different routes between source and destination. Agrawal and Mishra [17] presented a method to detect wormhole attacks in WSNs based on the RTT estimator for AODV protocol in Network Simulator-3 (NS-3).

3.1 Packet Leashes and Position of Nodes

The authors of [3] used packet leashes, which could be either geographic or temporal leashes, to limit a packet's maximum transmission distance. Finally, they presented the design and performance analysis of TIK, a novel, efficient protocol that also provides instant authentication of received packets, to implement temporal leashes. In a network of n nodes, 20 TIK requires only n public keys and has low storage, per-packet size, and computation overheads. In particular, a node only needs to perform 3 to 6 hash function evaluations per time interval to keep its key information up to date, and roughly 30 hash functions for each received packet. TIK has computational and memory requirements that are easily met today with commodity hardware such as 11 Mbps wireless links; 2.6 megabytes for hash tree storage, for example, represents less than 3% of the standard memory on a Compaq iPaq 3870 with no external memory cards, and since the StrongARM CPU on the iPaq is capable of performing 222,000 symmetric cryptographic operations per second, TIK imposes no more than an 18% load on CPU time, even when combined with precise timestamps and tight clock synchronization, TIK can prevent wormhole attacks, which cause the signal to travel a greater distance than the nominal range of the radio, or any other range that might be specified. Commercial GPS receivers can achieve sufficiently tight clock synchronization in a wireless LAN [21], and wireless MAN technology can be sufficiently time-synchronized using either GPS or LORAN-C [20] radio signals. A TIK-based MAC layer protocol effectively protects against replay, spoofing, and wormhole attacks while maintaining high freshness. Because the authentication of each packet can be performed on the host CPU, TIK is implementable with current technologies and does not require significant additional processing overhead at the MAC layer. Because they require broadcast authentication, geographic leashes are less

efficient than temporal leashes, but they can be used in networks where precise time synchronization is difficult to achieve. The ability to accurately measure location is the most important factor in the usability of geographic leashes; because node movement is very slow relative to the speed of light, the effects of reduced time synchronization accuracy are minor.

3.2. Location and Time-Based Approaches

Hu and Perrig [5] presented an approach using Packet Leashes, where geographic leash and temporal leash put an upper bound on the location of the receiver and the maximum time a packet takes to travel respectively. TIK protocol is proposed for defense against the temporal leash, but the knowledge of the geographic location or tight time synchronization is required. Taheri, Naderi, and Barekatin [23] used a leashes approach with a modified packet transmission methodology to decrease the calculation overhead of the TIK protocol. In the transmission time-based mechanism (TTM), Tran, Hung and Lee brothers [6] proposed an approach where each node on the path notes the time of sending the RREQ packet and receiving the RREP packet. Here, also time consideration is the main factor. Singh and Vaisala [14] modified this approach by removing the sender and receiver from maintaining request and reply packet timing. Hu and Evans [7] proposed a location-based approach, where a directional antenna is used to check the validity of neighbors. Considering the direction from which the response of the HELLO message comes and using verifiers, the neighbors are authenticated. The approach can detect insider attacks also by establishing authentication with pairwise secret keys, but hardware S M A B M D Close Wormhole Half-Open Wormhole Open Wormhole International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.5, Sep 2011 89 support is required here. Furthermore, this technology can only detect varieties of wormholes with false neighbors. For wormhole attack detection employing guard nodes, Khalil, Bagchi, and Shroff [4] presented a lightweight countermeasure (LITEWORP). After detecting the wormhole, LITEWORP leaves the network in that open mode only, causing the possibility of more disruption. To address this, they proposed MOBIWORP [8], a protocol that removes malicious nodes from networks using central authority, either locally or globally. Chen, Lou, Sun, and Wang [12] presented a secure localization approach that can detect simplex and duplex wormhole attacks. They extended this algorithm [10] to make it effective for dissimilar transmission ranges of sensor nodes also, but still multiple wormholes cannot be

detected by this. Nait-Abdesselam, Bensaou, and Taleb [11] proposed a detection and avoidance method that focuses on load-carrying by various routes. When a route is loaded heavily, it may be because of packet congestion, etc., so it may signal an alarm even when a wormhole is not present. Khurana and Gupta [12] proposed an approach based on the traveled distance and maximum transmission range of sensor nodes. SLEEP [12] was limited to the nodes with the same transmission range that has been extended as FEEPVR [11] to support dissimilar ranges also. Jakob Eriksson, Srikanth V. Krishnamurthy, and Michalis Fallouts [12] presented the True Link concept that has rendezvoused and authentication phases for wormhole detection. The former phase requires tight time synchronization, while the latter works on shared secret keys for signing messages.

3.3. Connectivity and Neighborhood-Based Approaches

Hayajneh, Krishnamurthy, and Tipper [2] presented a Secure Neighborhood (SECUND) protocol that can detect multi-ended wormholes. No need for specialized hardware, knowledge about the node locations, and no requirement of clock synchronization are positive points of this method, but it can work only if the presence of a wormhole increases fake neighbors by a considerable amount. Dimitriou and Giannetsos [15] derived an algorithm for wormhole detection based on connectivity information. The algorithm runs the local path existence test when it detects new nodes. Gupta, Kar, and Dharmaraja [21] presented an approach where the presence of a wormhole is found by the destination by counting the hop difference between the neighbors of one hop-away node. A special kind of Hand Packet is used for this purpose that introduces some processing delay also. Vani and Rao [22] proposed an approach WARRDP (Wormhole-Avoidance Route Reply decision packet) for wormhole detection and removal using the combined approach of Hop count, Anomaly-based, and Neighbor list methods.

K. Win [37] presented a security imitation based on the trust evaluation of nodes and neighbor monitoring. In this security model, sensor nodes go into immoral mode after sending the packet to neighbors. Subsequently, they observe the transmission status of RREQ packets. To analyze the correlation between packets sent and that dropped association coefficient has been made use of. The correlation coefficient is calculated for all the neighbors and the trust factor of a node is constructed. The vector containing the trust values of each of its neighbors is known

as the trust vector of a node. It is straightforward to detect the wormhole if the trust information is available through neighbor monitoring. During the routing stage, the algorithm for the detection of the wormhole is run.

3.4. Graphical and Topological Information-Based Approaches

Wang and Bhargava [16] presented a centralized approach MDS-VOW (Multi-Dimensional Scaling- Visualization of Wormhole) with the central controller. Here no hardware support is required, but it is less effective for the sparse network. A graph-theoretic approach was presented by R. Poorvendram and Lazos [17] that provides necessary and sufficient conditions to detect and defend against wormhole attacks. Specialized guard nodes, with high radio ranges, are the requirements of this methodology. Choi, Kim, Lee, and Jung [18] suggested a DSR-based Wormhole Attack Prevention (WAP) algorithm. The methodology works well for hidden assaults, but it is difficult to identify exposed attacks with this method. Vol. 3, No. 5, Sep 2011 90, International Journal of Network Security & Its Applications (IJNSA) Azer, Kassas, and Soudani [19] proposed a Diffusion of Innovations-based detection and prevention technique that works well except that the end-to-end delivery time is significantly increased.

3.5. Routing Algorithm-Specific Approaches

Poornima, Bindu, and Munwar [20] proposed a geographic routing scheme that detects the presence of wormholes using the Reverse Routing Scheme (RRS) and Authentication of Nodes Scheme (ANS). It is primarily applicable to the BSR protocol, and the value of the witness threshold is critical to the success of this approach. Attir, Abdesselam, Brahim, Bensaou, and Ben-Othman [6] proposed a method for detecting wormholes that use neighborhood detection, W-Delay, and appending additional information to the HELLO packet. This method works, but it is only applicable to the OLSR protocol.

3.6. Special Hardware-Based Approaches

In [4], a method was suggested in which sensor nodes are equipped with special directional antennas to defend against wormholes. They assume that if there is no wormhole attack and if one node sends packets in a given direction, then its neighbor will get that packet from the opposite direction. With a shared witness, the nearby nodes check the directions of the signals they receive from each other. The neighboring link is confirmed only when the directions of both pairs match. The disadvantages of using a directional antenna are each node is to be

equipped with special hardware. This method does not prevent multiple endpoint attacks. Directional errors are possible in this method.

3.7. Hop-Count Analysis Technique

In [4,] the author describes an end-to-end detection of a wormhole attack (EDWA) in a wireless ad hoc network. This method is divided into three stages. The first phase involves performing location-based end-to-end detection. The source detects wormholes here by estimating the shortest hop count between the source and destination. If the received shortest route's hop count is much lower than the estimated hop count, a wormhole is detected, and the source node sends an alert message to other nodes about the existence of a wormhole. The second phase involves wormhole identification. If there are multiple paths between the source and destination, the source node confirms the wormhole's endpoints using a TRACKING procedure. Once the endpoints are identified, the results are broadcasted throughout the network to alert other nodes to the presence of malicious nodes. Finally, in the final phase, a genuine route for data communication that is legitimate and free of wormholes is chosen. As a result, this approach serves as both a detection and identification method, with no special hardware requirements or clock synchronizations required. However, this proposed method is only effective when the source and destination are not too far apart.

CHAPTER 4

DESIGN OF THE PROPOSED SOLUTION

This Chapter is all about the proposed system. The proposed system is a security approach to detect and mitigate wormhole attacks. It is a secured AODV approach that efficiently detects wormhole nodes present in a Wireless Sensor Network and prevents them by removing wormhole nodes from the network. It calculates the distance from source to destination and computes the minimum number of routing hops that will be in the network during communication between source and destination. After that, it decides the presence of a wormhole attack in the network if the actual routing hop is less than the minimum number of routing hops computed from their respective distance within the given maximum transmission range of one node. Based upon this hop count method it decides whether the network is exposed to wormhole attacks or not. Afterward, explore which nodes transmitted above the maximum transmission range and remove that malicious (wormhole) node from the network (drop route request or route replay message from the detected wormhole nodes).

It is one of the secured solutions because it uses the hop count method to detect and prevent wormhole attacks. To detect wormhole attack in the proposed system, the maximum transmission range and the minimum number of hops at that range is used. to calculate the distance from the source to the destination it used the Pythagoras distance formula [39].

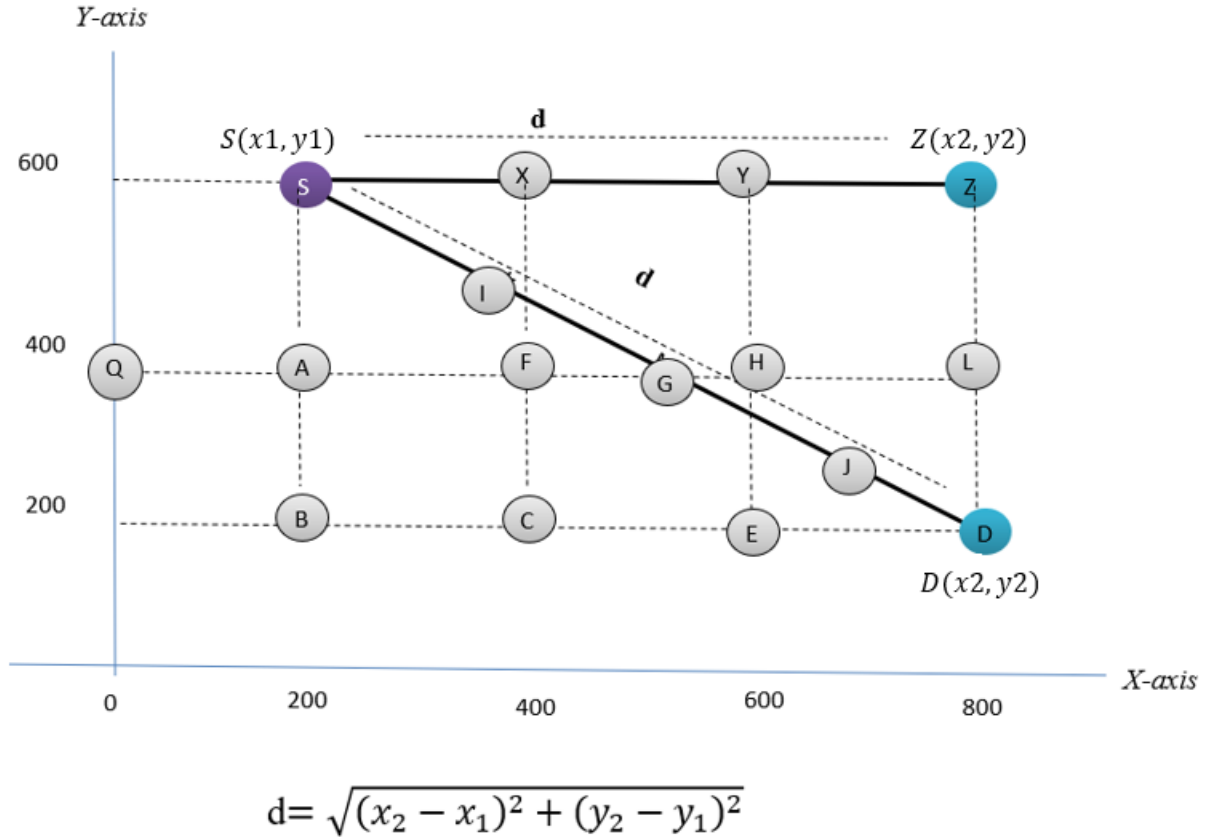


Fig 4. 1 location of nodes in a 2D graph

where x_1 and y_1 is the position of the source node and x_2 and y_2 is the position of the destination node.

In general, if there are n points in a straight line or colinear, the minimum number of line segments is $n-1$. In this case, the number of points is the number of nodes they are in the route from source to destination and the number of line segments is the number of routing hops from source to destination.

Once the distance from the source to the destination is known, we can derive the number of hops and number of points (nodes) on the route because the maximum length of one segment (routing hops) is known. In our method we have used the maximum transmission range of one node is 200m, so if the distance from source to destination is greater than this range (200m), simply there must be another intermediate node between them (source and destination).

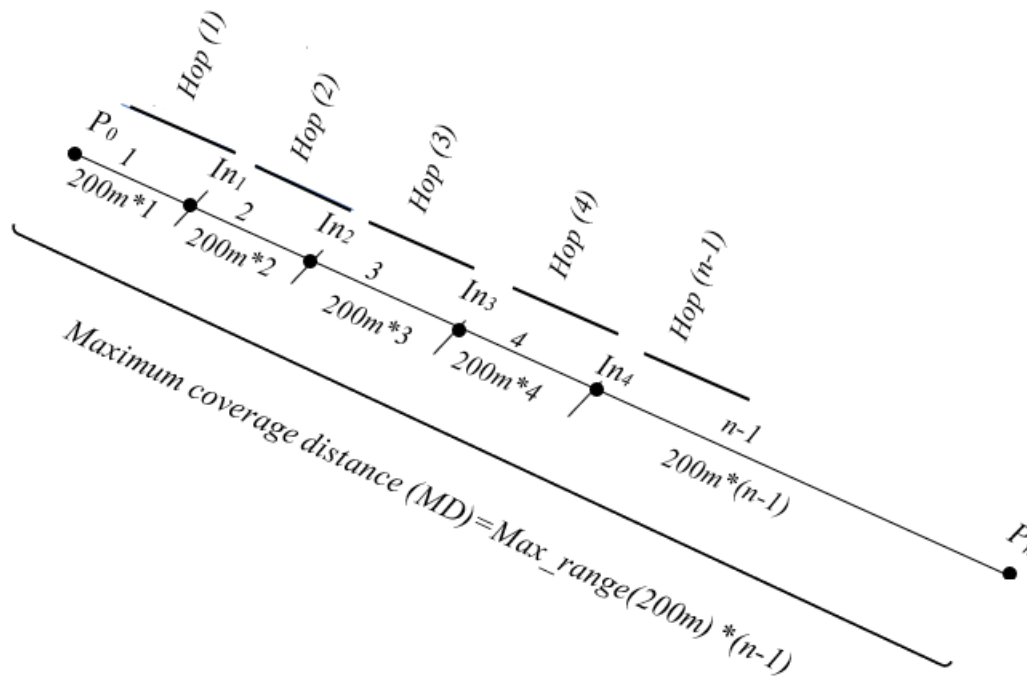


Fig 4. 2 distance between nodes on fixed transmission range

Where P_0 and P_n are the sources and destination nodes

$In_1, In_2, In_3,$ and In_{n-1} are Intermediate nodes between the source and destination when the route is established.

200m is the maximum transmission range of each node.

The number $1,2,3$ up to $n-1$ represents the number of line segments (number of hops) from the source node to another node.

The important formula to find the distance from the source to the destination is

$$D \leq (n - 1) * (max_range) \text{ hence } d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \text{ ----- (Equation 4.1)}$$

$$\text{From this formula, we can get } (n - 1) \geq (\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}) / max_range$$

Where $(n - 1)$ is the number of routing hops (no of hops)

int this method we used the maximum transmission range within 200 meters and to find the minimum number of routing hops in this range we have to divide the distance calculated from source to destination by the maximum transmission range ($d/200$).

From the above diagram number of hops from source to destination are equal to the number of line segments from source to destination.

$$No_hops = d / Max_range \text{ -----(Equation 4.2)}$$

$$No_hops = (\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}) / 200 \text{ -----(Equation 4.3)}$$

if we get a floating-point it ceils to the next largest integer digit.

$$No_nodes = No_hops + 1 \text{ -----(Equation 4.4)}$$

4.1 Architecture of Detection and prevention of Wormhole Attacks

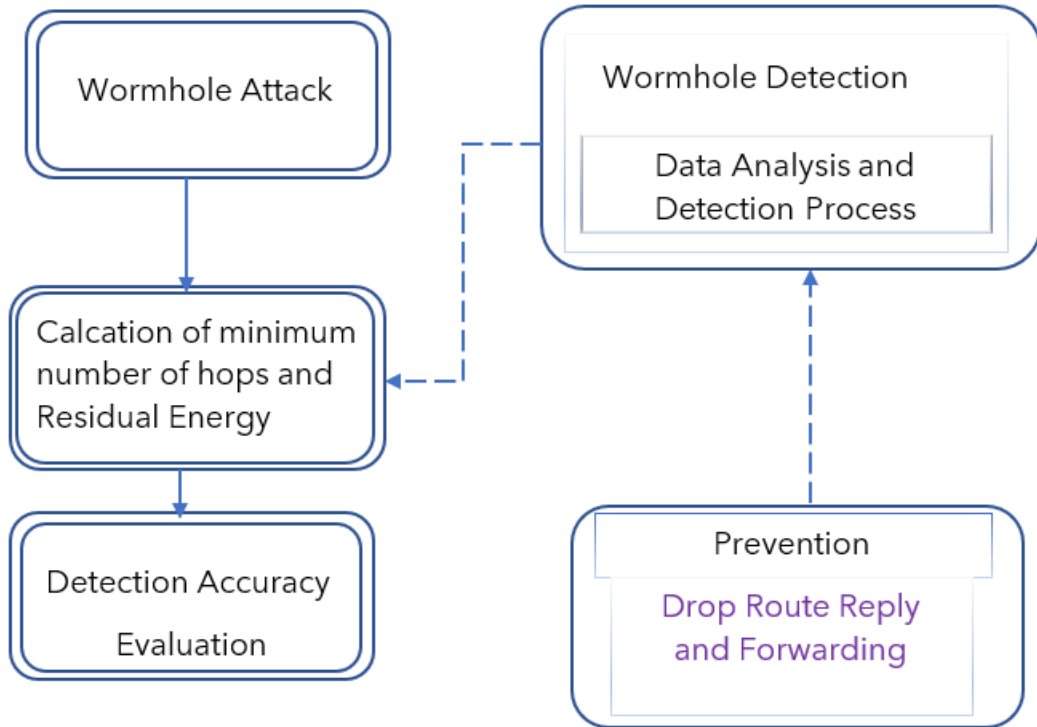


Fig 4. 3 Architecture of wormhole attacks detection in WSN

A) Wormhole Attack

There number of sensor nodes present in a network. Wormhole nodes can be structured using out-of-band channels, where two wormhole nodes are explicitly combined with a long range of wireless links. The main purpose of wormhole attacks is to gain sensitive information from the network.

B) Calculation of minimum number of hops and Residual Energy

Calculation of the minimum number of hops can be derived from the distance from source to destination dividend by the maximum transmission range of nodes to detect the presence of wormhole attack in a network. Detection of wormhole attacks is based on the following three steps.

- I. The source node sends RREQ packets to receiver nodes in the network, and when the receiver receives RREQ, it replays through RREP.
- II. The source node checks the actual number of hops received from receiver nodes, Tracks the location of each node using GPS in the network, and calculates the minimum number of hops (source node to each node in the network).

$$MnH = SD/M \text{ and ceil to the next integer.}$$

Where MnH is the minimum number of hops from source to destination, SD is the distance from source to destination and M is the maximum transmission range (200m) of nodes

- III. Finally compute the residual energy of each node, $Initial\ energy - Consumed\ energy$.

$$Initial\ Energy = 50\ joules$$

C) Data Analysis and Detection

By checking the minimum number of hops from source to destination, compare it with the actual routing hops (AnH). $If (AnH < MnH)$, wormhole presence in the network, and identity of which one is the first wormhole. Detection of the first wormhole is based on the following steps.

- I. List all active nodes in the network and sort based on MnH
- II. Add all active nodes to nx3 matrices, the first, second, and third columns contains Node-ID, MnH and Residual Energy.
- III. Check if nodes have neighbors or not,

- ⇒ If it has neighbors, remove it from the matrix
- ⇒ Else record as the first wormhole.

Second wormhole detection is based on the following steps.

- I. List all active nodes found in the matrix above the first wormhole transmission range.
- II. Sort based on the distance from the destination
- III. Check whether a node has neighbors or not,
 - ⇒ If it has neighbors, remove it from the matrix
 - ⇒ Else if nodes haven't neighbors, and the number of nodes that haven't neighbors is one, record it as a second wormhole.
 - ⇒ If the number of nodes hasn't neighbors is more than one node, and more than one node have equal MnH , record the second wormhole when nodes have greater MnH
 - ⇒ Otherwise check the residual energy of each node those have equal MnH and record the second wormhole when a node has less residual energy.

D) Drop RREP and Forwarding

Since WSNs are vulnerable to various attacks, it is necessary to protect a network from different types of attacks. One of the most common attacks in WSN is wormhole attacks and we tried to prevent or isolate wormhole nodes from a network by adding wormhole nodes to blacklists, dropping RREP comes from when nodes are in blacklists, and forwarding data packets to the normal nodes.

E) Detection Accuracy Evaluation

Evaluation of the detection accuracy of our methods depends on the successful detection of wormhole nodes from the total number of adversary wormhole attacks carried out in the network.

$$DR = NSD/TNA$$

Where DR , Detection Rate, NSD , Number of Successful Detection, and TNA , Total number of Attacks

The Wormhole detection and Prevention AODV approach are implemented in the following steps

Step 1: Source Initialization: Initialize the source in WSNs using the AODV protocol.

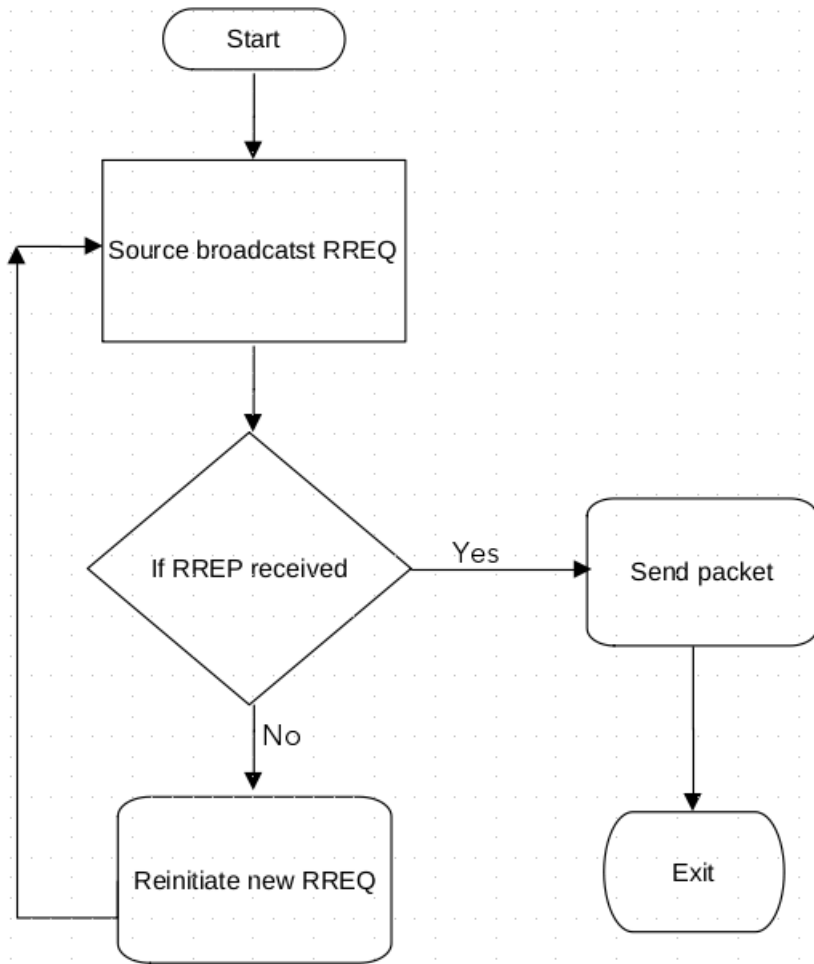


Fig 4. 4 Source node route initialization

Step 2: Detection of wormhole attack takes place based on the above flowchart:

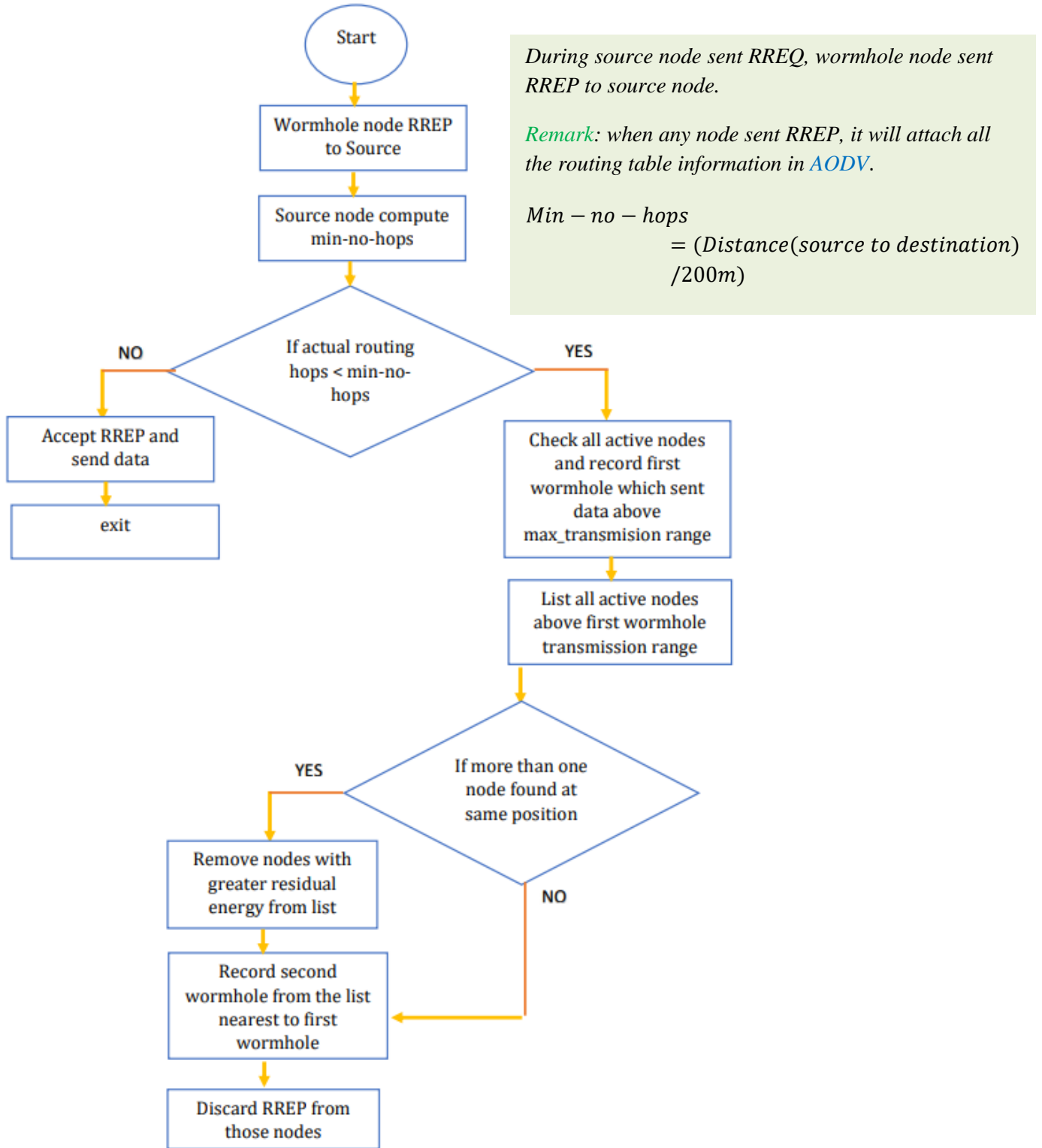


Fig 4. 5 Wormhole detection and prevention flowchart

❖ Assumptions

In this section, some assumptions are presented regarding network and opponent capabilities in the proposed design in WSN.

- ⇒ Assumption 1: two nodes are considered neighbors, if the distance between them is within the transmission range (200m).
- ⇒ Assumption 2: the nodes start with the same energy level and the attacker node has a random speed and mobility direction.
- ⇒ Assumption 3: malicious nodes can launch out-of-band wormhole attacks.
- ⇒ Assumption 4: All sensor nodes are statically deployed in the two-dimensional square network.

- ⇒ Assumption 5: Nodes are equipped with GPS to determine their location

4.2 Out-Of-Band Wormhole Detection

I. Transmission Range Phase

To illustrate this phase of the proposed algorithm, it's important to identify the nodes within its communication range for each network node. This phase relies on the hop count method based on the distance from source to destination and the maximum transmission range for two successive nodes to conclude the presence of a wormhole attack in the network. to calculate the minimum number of routing hops in the network we will divide the distance from source to destination by the maximum transmission range between two successive nodes. The nodes that are not in range of the source node will be considered malicious nodes, due to limited radio coverage and the distribution of the legitimate nodes which are closer to one another.

Algorithm 1:

Out-of-Band wormhole attack creation

Taking after strides are proposed to detect and prevent wormhole attacks in WSN:

Step1: Creation of Sensor Network Scenario

Step2: Deployment of first Wormhole node utilizing High Transmission Power Sensor Node anywhere in the network and deploy the second wormhole everywhere outside the transmission range (200m) of nodes from the first wormhole.

Step 3: Increase Transmission Power esteem for both wormholes.

Step 4: Modify AODV to misroute and drop all got parcels

Step 5: Detection of Wormhole attack by checking Transmission range, hop count and Residual energy.

Algorithm 2:

Out-of-Band wormhole attack Detection Algorithm

Input: Transmission range value, routing hop value.

Output: wormhole attack detected

1. Start
2. Nodes are deployed using AODV protocol
3. Calculate the minimum routing hop MnH by dividing the distance from source to destination D by the maximum transmission range for one node, MAX_RANGE .

$$MnH = D/MAX_RANGE \text{ and ceil to the next larger integer if a floating point is found}$$

4. find the actual routing hop from route table AnH .
5. If $(MnH \geq AnH)$ then
6. A neighboring node in the range of source node
7. No Out-of-band wormhole detected, go to algorithm 4
8. else

9. Out of band detected, Add Malicious node to Blacklist to and
10. Prevention (drop RREP from this node).

CHAPTER FIVE

IMPLEMENTATION AND RESULT EVALUATION

5.1 Overview

In this work, we assume WSNs are homogeneous (all network nodes contain the same hardware and software configuration), symmetric (node A can only communicate with node B if and only if B can communicate with A), and static (network nodes can't move after deployment). In particular, the radio transceivers of all members of the network operate under the same configuration throughout the lifetime of the network (e.g., transmission power, antenna height, and antenna gain).

All nodes are uniquely identified and know their geographical position, which can be obtained using a positioning system such as the GPS. The value of a node's geographical position, as well as its identifier, are included in each of the messages it sends.

A wormhole attack is one of the gravest attacks that are considered a challenging problem and can be launched at the network layer of the OSI model [25]. It consists of two malicious nodes involved in the routing path and communication links between them as illustrated in Fig.1.1 between two wormhole nodes.

This chapter deals with the implementation of the proposed solution in a simulation environment and the tools we have used during the prototype implementation of the proposed system are described in detail. The performance and confidentiality of the proposed system are evaluated and compared with AODV with modified AODV using performance measuring metrics and analytical discussion on the results from our point of view. The goal of the simulation is to design, simulate and analyze the confidentiality, availability, and integrity of the proposed system by comparing it with AODV using security and performance metrics. The empirical study of the performance results is analyzed from experimental analysis using the trace file generated during simulation run time. This trace file contains the events which occur during communication between nodes when we run the system.

The general conclusion on the performance is drawn from the analytical observations obtained from the simulation. The proposed system is evaluated by using various security and performance metrics such as packet delivery ratio, end-to-end delay, throughput, jitter, and packet loss to evaluate the effectiveness whether achieving the objectives.

5.2 Simulation Tools and Development Languages

In this Section simulation environment and development languages that we used in the implementation and evaluation of the proposed solution are described in detail. It is known that simulation plays an important role in sensor Wireless Sensor Networks to design, implement and evaluate real-world communication scenarios that help us to evaluate the security and performance of the communication system scientifically. It is an important technique used to realize and show how the real-world communication system operates. Simulation is widely used in exploring and modeling different communication systems for many application areas like military applications, education, healthcare, environmental monitoring (i.e., earthquake), etc. It is used to deploy real network communication systems as well as reduce the cost of building and testing any proposed model by doing experimentation. There are lots of different network communication simulators that have been developed with their powerful features that cover different characteristics of WSN. Among the major network, communication system simulation tools are NS-2, GLOMOSIM, OMNET++, QUALNET, J-SIM, OPNET, and NS-3.

All these aforementioned simulation tools have their characteristics that should be considered to make a simulation for the WSNs environment. Therefore, selecting the proper simulator by assessing which one will provide optimum performance and suitability of network for implementing and evaluating the proposed work is the critical factor that should be considered in the simulation of many communication systems, especially in WSNs. According to the survey which has been made on those simulation tools, NS-2 is better in the simulation of wireless sensor networks and we select this for our simulation purpose.

5.2.1 OMNET++

⇒ Objective Modular Network Testbed in C++ and this is the abbreviation of OMNeT++.

⇒ It is a simulator that is specially designed for the discrete events in distributed systems and also it is extensible, open-source software, modular and extensible.

⇒ This simulation includes various atomic behaviours of simplex models.

✚ Supported Operating systems

⇒ MAC OS, Linux, UNIX, Windows (XP, Win2K)

✚ Features of OMNeT++

⇒ It is a good structures simulator like NS2

⇒ C++ programming language is used to construct distributed systems and communication networks.

⇒ MAC-based scheduling, localization, and routing can be implemented for different wireless networks.

5.2.2 NS-3 (Network Simulator-3)

Like NS-2, NS-3 is also regarded as a discrete-event simulator. The objective behind its development was to enhance research in communication networks. NS-3 is an open-source simulator and was launched in June 2008. The latest version is 3.21 released in August 2014. NS-3 is not regarded as an extension of the NS-2 simulator, NS-3 is a new simulator not supporting any APIs belonging to NS-2.

As programs written in NS-2 are coded in OTcl and results can be visualized using NAM and XGraph but pure C++ code is not possible in NS-2. But in NS-3, all the programs are written in pure C++ with optional python bindings.

There is no Graphical Tool which is available in NS-3, but still graphical results can be interpreted using NetAnim open-source software.

5.2.3 GLOMOSIM

GloMoSim (Global Mobile Information System Simulator) is discrete event scalable simulation software that simulates wireless and wired network systems.

5.2.4 NS-2(Network Simulator-2)

NS2 is an open-source simulation tool that runs on Linux. It is a discreet event simulator targeted at networking research and provides substantial support for the simulation of routing multicast protocols, and IP protocols, such as UDP, TCP, RTP, and SRM over wired and wireless (local and satellite) networks.

- ⇒ Widely known as NS2, is simply an event-driven simulation tool.
 - ⇒ Useful in studying the dynamic nature of communication networks.
 - ⇒ Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2.
 - ⇒ In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors.
-
- ❖ Due to portability (TCL code written in NS-2 is can be imported in NS-3), low CPU usage, and support of built-in GUI, we selected NS-2 for testing our proposed methods.
 - ❖ On the other hand, the NS-2 trace file can be easily interpreted by writing any programming language and is easy to understand.
 - ❖ NS2 provides emulation functionalities and can be used for parallel and distributed simulation

Protocols implemented in NS2

- ⇒ Transport layer (Traffic Agent) – TCP, UDP
- ⇒ Network layer (Routing agent)
- ⇒ Interface queue – FIFO queue, Drop-Tail queue, Priority queue
- ⇒ Logic link control layer – IEEE 802.2, AR

5.2.5 Trace data analyzing applications in NS-2

The applications used for analyzing trace files produced from the simulation are XGraph, NS2-VisualTraceAnalyzer, NsGTFA, TraceGraph, and AWK.

5.2.5.1.X-Graph: The graph is used for lucrative plotting and generating graphs. To use XGraph in NS-2, it should be called within a TCL Script. It will load a graph showing the visual information of the trace file produced in the simulation.

5.2.5.2 TraceGraph: Trace graph is a good application that comes very easily to NS2 users. It removes the coding of awk scripts that are needed to configure and run over the trace file. Trace graph makes analysis very simple. Trace graph seems to have been developed using MATLAB and therefore supporting codes are needed to make it run in Linux.

5.2.5.3 AWK: To process and extract important information from a huge amount of data or text, a scripting language is essential. From those scripting languages for our research, we prefer to use the AWK programming language because it is suitable for processing a huge amount of network trace files to produce performance results.

5. 3 Network Components in a Sensor Node

The network stack for a sensor node consists of a link layer (LL), an ARP module connected to LL, an interface priority queue (IFC), a MAC layer (MAC), and a network interface (netIF), all connected to the channel. These network components are created and plumbed together in Tcl. Each component is discussed briefly as follows.

Link Layer

The LL object is responsible for simulating the data link protocols like in wired networks. The only difference being the link layer for the sensor node has an ARP module connected to it which resolves all IP to hardware (Mac) address conversions. Normally, the Routing Agent goes all inflow and outflow (into channel) packets to the LL. The LL routes the data packet to the interface queue. The mac layer forwards all incoming packets (out of the channel) to the LL, which then forwards them to the node entry layer. CMU has implemented the IEEE 802.11 distributed coordination function (DCF) Mac protocol. It uses an RTS/CTS/DATA/ACK pattern for all unicast points. [35]

ARP

The Address Resolution Protocol module receives queries from the Link layer. If ARP has the hardware address for the destination, it writes it into the Mac header of the packet. Otherwise, it broadcasts an ARP query and temporarily caches the packet. For each unknown destination hardware address, there is a buffer for a single packet. In case, additional packets to the same destination are sent to ARP, the earlier buffered packet is dropped. Once the hardware address of a packet's next hop is known, the packet is inserted into the interface queue. [35]

Interface Queue

The PriQueue. /ns-2/priqueue.h [35] is implemented as a priority queue which gives priority to routing protocol packets, inserting them at the head of the queue. It supports running a filter over all packets in the queue and removes those with a specified destination address. [35]

Mac Layer

The IEEE 802.11 distributed coordination function (DCF) Mac protocol has been implemented by CMU. It uses an RTS/CTS/DATA/ACK pattern for all unicast packets and simply sends out DATA for all broadcast packets. The implementation uses both physical and virtual carrier sense. [35].

Network Interfaces

The Network Interface layer serves as a hardware interface that is used by a sensor node to access the channel. The wireless shared media interface is implemented as Phy/WirelessPhy./ns-2/wireless-PHY.h [35]. Collisions occur on this interface, and the radio propagation model receives packets sent by other node interfaces to the channel. Each transmitted packet is stamped by the interface with meta-data related to the transmitting interface, such as transmission power and wavelength., etc. This meta-data in the packet header is used by the propagation model in receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected (carrier sense) by the receiving node. The model approximates the DSSS radio interface (Lucent WaveLan direct-sequence spread-spectrum). [35]

Radio Propagation Model

It employs Friss-space attenuation () at close ranges and an approximation to Two ray Ground () at long ranges. The approximation assumes a specular reflection of a flat ground plane. See tworayground. {cc,h}in [35].

Antenna

An Omnidirectional antenna having unity gain is used by sensor nodes. See antenna. {cc,h} for implementation details in[16].

Routing Protocols or Agents

There are four different types of Ad-hoc routing agents defined by NS-2 currently.

These are: -

- ⇒ AODV (Ad-hoc On-Demand Distance Vector)
- ⇒ DSDV (Destination Sequenced Distance Vector)
- ⇒ DSR (Dynamic Source Routing)
- ⇒ TORA (Temporal Ordered Routing Algorithm)

NS-2 Trace Support

The NS-2 trace files are used for post-processing the ongoing simulation. The following trace file formats are supported:

- ⇒ Trace files for wired networks that are wired
- ⇒ Satellite
- ⇒ Wireless (old and new trace)
- ⇒ Wired-cum-wireless

Currently, cmu-trace objects are used for tracing in wireless simulations. In the simulation's future, this will be expanded to include trace and monitoring support from NS2, as well as NAM support for wireless modules. There are three types of cmu-trace objects: CMUTrace/Drop, CMUTrace/Recv, and CMUTrace/Send. These are used in NS2 to trace packets dropped, received, and sent by agents, routers, mac layers, or interface queues.

5.4. Structure of Trace Files

| Event | | Time | From Node | To Node | Pkt Type | Pkt Size | Flags | Fid | Src Addr | Dest Addr | Seq Num | Pkt ID |
|-------|--|------|--------------|------------|-------------|-------------|-------|-----|-------------|--------------|------------|-----------|
|-------|--|------|--------------|------------|-------------|-------------|-------|-----|-------------|--------------|------------|-----------|

1. The first field is the event type. It is given by one of four possible symbols r, +, -, and d which correspond respectively to receive (at the output of the link), enqueued, dequeued, and dropped.
2. The second field gives the time at which the event occurs.
3. Gives the input node of the link at which the event occurs.
4. Gives the output node of the link at which the event occurs.
5. Gives the packet type (e.g., CBR or TCP)
6. Gives the packet size (number of packets that can be sent)
7. Some flags (route request, route replay, and errors report, etc...)
8. This is the flow-id (fid) of IPv6 that a user can set for each flow at the input OTcl script one can further use this field for analysis purposes; it is also used when specifying stream color for the NAM display.
9. This is the source address given in the form of a node port.
10. This is the destination address, given in the same form.
11. This is the network layer protocol's packet sequence number. Even though UDP implementations in a real network do not use a sequence number, ns keeps track of UDP packet sequence numbers for analysis purposes
12. The last field shows the Unique id of the packet.

5.5. Implementation Details

AODV is generally efficient and scalable in terms of network performance, but it allows attackers to easily advertise falsified route information to redirect routes and launch various types of attacks. Some critical fields in each AODV routing packet, such as hop count, source and destination sequence numbers, IP headers, AODV source, and destination IP addresses, and RREQ ID, are required for proper protocol execution. Any abuse of these fields can cause AODV to fail.

| Parameters | Value |
|-----------------------------------|----------------|
| Area | 1600X702 |
| Simulation Time | 10 seconds |
| N _o of Nodes | 10,32 |
| Traffic Model | FTP, CBR |
| Traffic type | TCP, UDP |
| Routing Protocol | AODV |
| Transmission Range | 200m |
| N _o network connection | 1/2/3/ |
| Mac protocol | 802.11 |
| Packet size | 512 |
| Propagation Model | Tow ray ground |
| N _o of wormhole nodes | 2 |
| Initial energy | 50.0 joules |
| Transmission power | 1.0 watts |
| Receiving power | 0.5 watts |
| Sleep and idle Power | 0.1 watts |

Table 5. 1 simulation parameters

One of the important things to know is the presence of malicious (wormhole nodes), it is better to check the routing table information that is maintained during the route established from source to destination.

| <i>Source Node</i> | <i>Route established time</i> | <i>Destination Node</i> | <i>Next hop</i> | <i>Number of hops</i> | <i>Route Sequence number</i> | <i>Route expired time</i> |
|--------------------|-------------------------------|-------------------------|-----------------|-----------------------|------------------------------|---------------------------|
|--------------------|-------------------------------|-------------------------|-----------------|-----------------------|------------------------------|---------------------------|

Table 5. 2 route table format

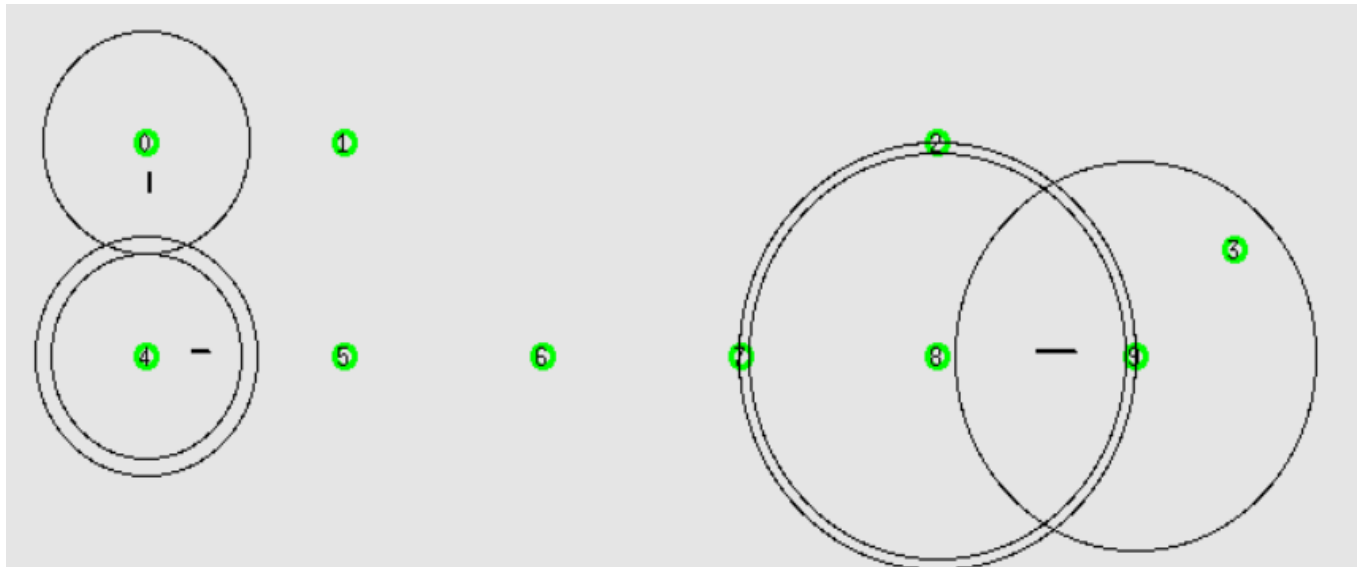


Fig 5. 1 AODV without wormhole simulation

In this simulation 0 is the source node, 9 is the destination node and 4 and 1 are the next hop for a certain time

| <i>source node</i> | <i>started time</i> | <i>destination node</i> | <i>next hop</i> | <i>number of hops</i> | <i>sequence number</i> | <i>route expired time</i> |
|--------------------|---------------------|-------------------------|-----------------|-----------------------|------------------------|---------------------------|
| 0 | 0.052545 | 9 | 4 | 6 | 4 | 10.052545 |
| 0 | 1.5 | 9 | 1 | 6 | | 11.5 |
| 9 | 5.054929 | 0 | 8 | 6 | 20 | 15.054929 |

Table 5. 3 route table without Wormhole attack

in this table, the actual routing hops are 6 and the minimum routing hops are 6 so, there are no nodes transmitting data above the transmission range.

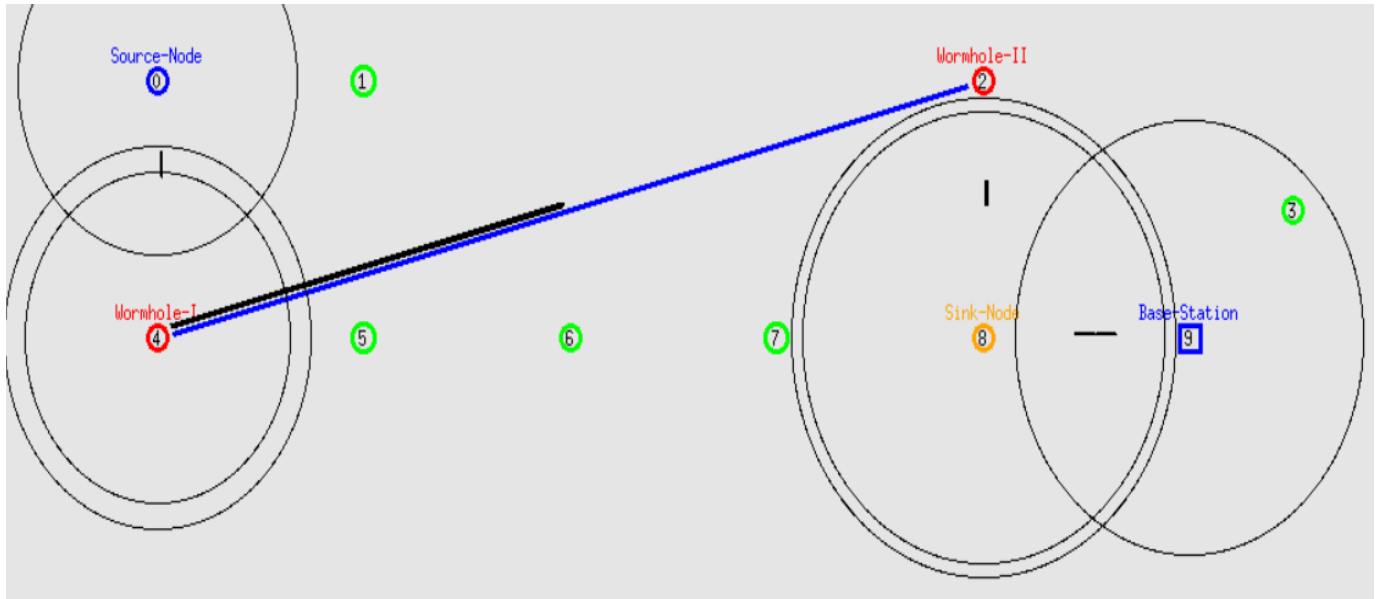


Fig 5. 2 AODV with wormhole simulation

In this simulation 0 and 9 are the source and destination, 4 and 2 are the first and second wormhole

in this scenario, the distance from source to destination is 1019.8 and the number of actual routing hops is 4. but the routing hops from source to destination must be greater than the distance from source to destination 1019.8 divided by the maximum transmission range of nodes (200). so, the minimum routing hops are $1019.8/200 = 5.099 \sim 6$. In this case, if routing hops are n , then the number of nodes on the route is $n + 1$. as Fig 5.4 and Table 5.4 shows that the number of hops is 4, which means the number of active nodes in the route is $n + 1 = 5$. Two of them are source and destination, and the other two nodes are wormhole node 4,2 and the other one is Sink node 8.

| Source node | Started Time | Destination Node | Next hop | Number of hops | Sequence Number | Route expired time |
|-------------|--------------|------------------|----------|----------------|-----------------|--------------------|
| 0 | 0.016872 | 9 | 4 | 4 | 4 | 10.016872 |

Table 5. 4 route table with wormhole

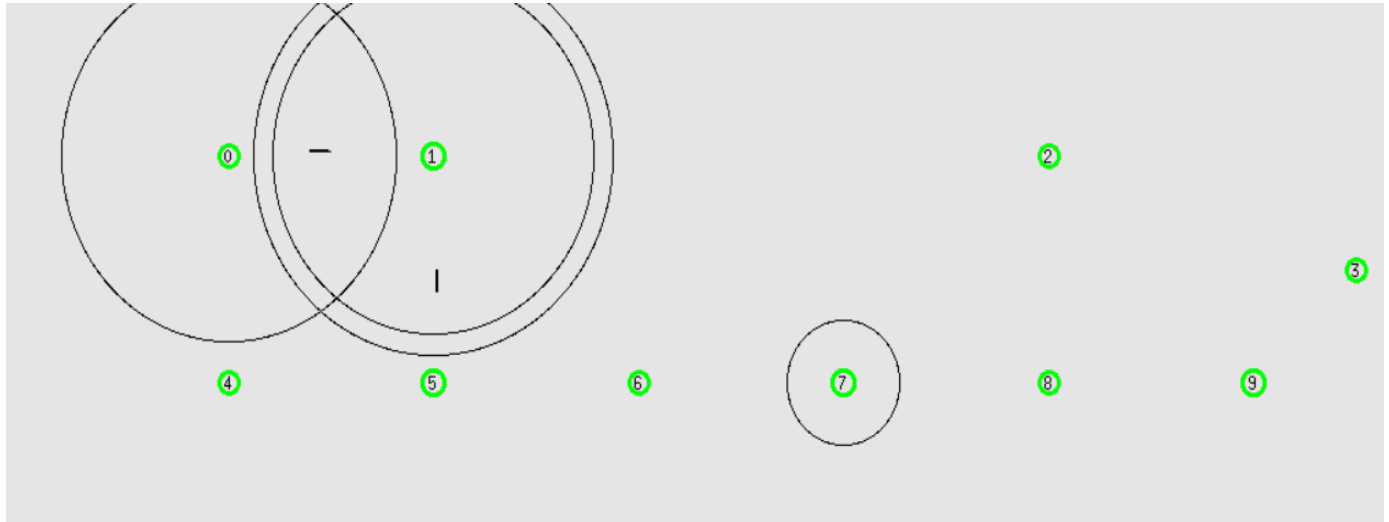


Fig 5. 3 AODV with WDPT simulation

Fig 5.5 shows that after detecting wormhole nodes in the network, all nodes reject route replay from those wormhole nodes that are recorded in the blacklist.

since the two wormhole nodes are detected, each node in the network rejects the route reply from those malicious(wormhole) nodes. in this case, the wormhole nodes are out of the network and simply they can't launch an attack on the network.

| Source node | Started Time | Destination Node | Next hop | Number of hops | Sequence Number | Route expired time |
|-------------|--------------|------------------|----------|----------------|-----------------|--------------------|
| 0 | 0.053055 | 9 | 1 | 6 | 4 | 10.053055 |
| 9 | 3.549188 | 0 | 8 | 6 | 22 | 13.549188 |

Table 5. 5 WDPT route table

Before applying the wormhole detection and prevention method, Fig 5.6 shows that wormhole node 4 average throughputs received from source node 0.

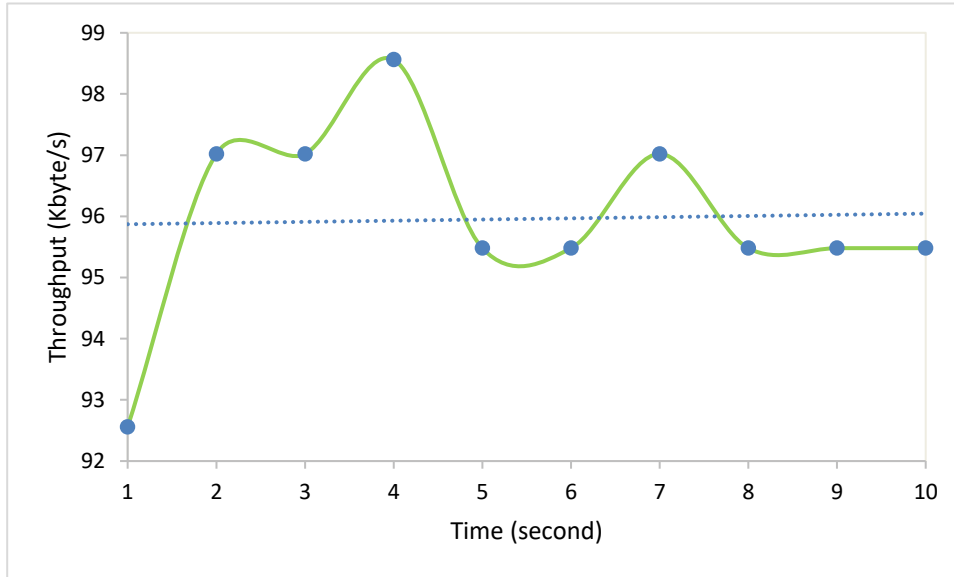


Fig 5. 4 Throughput Received by node 4 from node 0

As fig 5.6 shows that by misrouting source node0, wormhole node4 received nearly 96 Kbyte/s of data within 1 second time interval (average throughput). During this attack, another neighbor's node (node1) has the same number of hops to the destination. but the average throughput of neighbor node 1 is 0. It shows that wormhole nodes could make high traffic attraction and misroute source node 0 throughout the network lifetime.

Fig 5.7 shows that After applying the wormhole detection and prevention method, neighbor node 1 average throughput received from source node0.

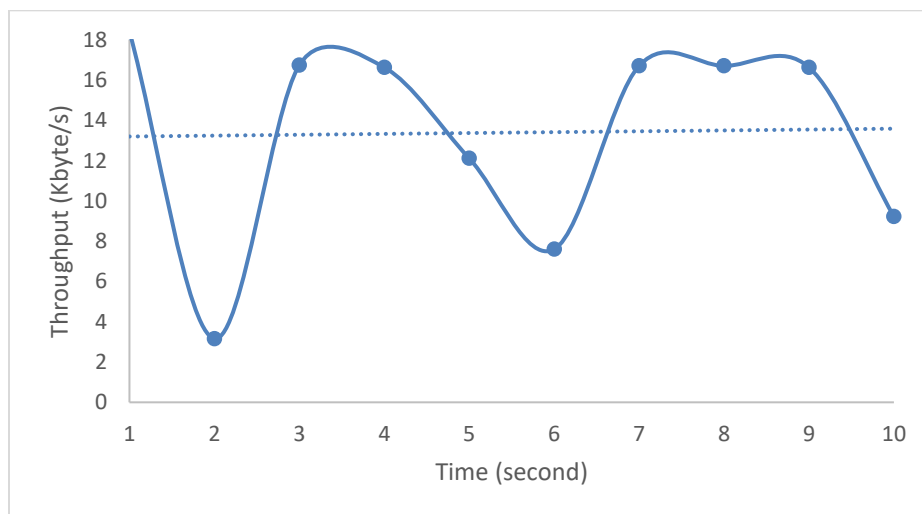


Fig 5. 5 Throughput Received by node 1 from node 0

After applying the wormhole detection and prevention method the two wormhole nodes involved in the wormhole attack throughput are 0. This means all nodes reject the route reply from those malicious nodes that are recorded in the blacklist. Fig 5.7 shows the throughput received by node1 from source node0. since node1 is a normal node, all of the packets are transferred through it due to there is no nearest node to source node0.

Fig 5.8 shows that average throughput of neighbors (node 1 and node 4) received from source node 0 without a wormhole attack.

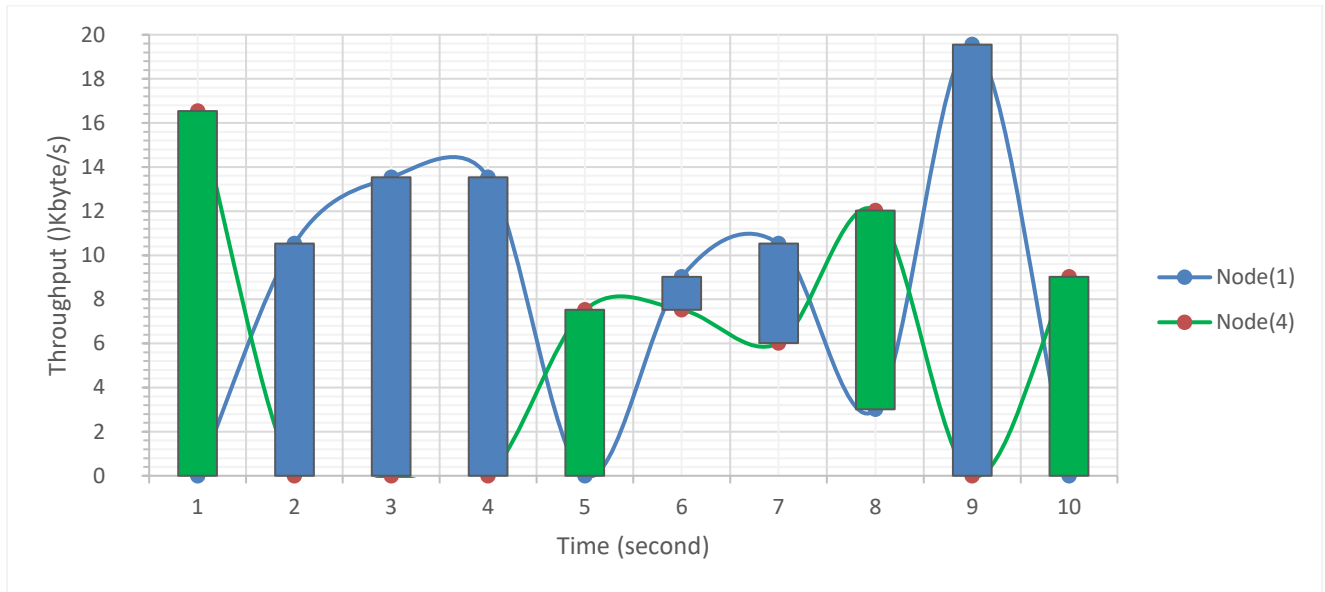


Fig 5. 6 Throughput received by node1 and node4

Fig 5.8 illustrates that comparison of neighbor nodes throughput received between 1 second time interval (1000 Milliseconds), as we see from the diagram when starting simulation until 1-second node4 throughput is 16.54 and node1 is 0, from 2 – 4 seconds node1 throughput is not 0. This means for that time 1 is elected as the next hop for source node 0 and until 5 seconds node4 is elected as the next hop and 6-8 both node1 and node4 are elected as the next hop for source node 0. Finally, until 9 seconds node 1 is elected as the next hop, and after 9-10 seconds node 4 is elected as the next hop.

Table 5.6 shows the average residual energy of the networks with wormhole attack, without wormhole attack, and with wormhole detection and prevention (WDPT).

| | <i>Normal</i> | <i>Attack</i> | <i>WDPT</i> | |
|-------------------------------|---------------|---------------|-------------|-------------------|
| <i>Initial energy</i> | 50 joules | 50 joules | 50 joules | Simulation time |
| <i>Energy consumption</i> | 6.764064 | 9.06581 | 6.14664 | 10 <i>seconds</i> |
| <i>GPS energy usage</i> | 0 | 0 | 3.308 | 10 <i>seconds</i> |
| <i>Residual energy</i> | 43.235936 | 40.93419 | 40.5453565 | 10 <i>seconds</i> |

Table 5. 6 Average Residual Energy

GPS energy usage we used in this paper is taken from Thomas Graf Berlin University of Technology 2011/12 conducted on smartphones [40][41].

In [40][41], the authors stated that to the position of nodes for one minute 6.616 joules of energy are needed. On the other hand, some experiments show that a GPS signal can reach from a satellite to earth below (1/15) second [43]. Due to this for sending and receiving signals we used 30 seconds.

To get the average residual energy of nodes, we run a simulation based on the following criteria.

- ⇒ Without wormhole nodes 1 times
- ⇒ With wormhole nodes 19 times
- ⇒ And with wormholes and WDPT 19 times

The number 19 implies the possible number of wormhole attacks for the 10 nodes that we used in our simulation.

$$\mathbf{Residual\ Energy} = (\mathbf{Initial\ Energy} - \mathbf{Energy\ consumption}) - \mathbf{GPS\ energy\ usage}$$

$$\mathbf{WDPT\ Residual\ Energy} = (50 - 6.14664 - 3.308)$$

$$= \mathbf{40.5453565}$$

- The most important thing for WSN is knowing the exact location of nodes for finding the shortest route for sending and receiving gathered data to the base station. As we have seen from table 5.6 energy usage of WDPT is high because of GPS but finding the exact location of nodes is critical for WSN to know which nodes are reachable or not.
- Table 5.7 illustrates the average throughput and packet delivery ratio of the network without the wormhole, with the wormhole, and with WDPT.

| | Normal | Attack | WDPT |
|--------------------------|---------------|---------------|-------------|
| <i>Generated Packets</i> | 97 | 632 | 100 |
| <i>Received packet</i> | 92 | 621 | 86 |
| <i>PDR(bytes)</i> | 94.85 % | 98.26 % | 86 % |
| <i>Throughput(KbPs)</i> | 0.076123 | 0.497250 | 0.068811 |

Table 5. 7 Throughput and PDR

As we see from Table 5.7, the number of packets generated is six times greater than from the normal network and WDPT. It causes the network unstable and buzzy for replying to these wormhole nodes' requests instead of sensing or gathering required data and sending it to the base station. The number of throughputs is also higher than normal and WDPT, but it is not usable for that network because most of the packets are generated by malicious nodes, only wormhole nodes use it.

Table 5.8 describe the impact of wormhole attacks on AODV routing protocol before simulating our method. We sent 632 packets in 6 hops, and 621 packets in 10 hops and monitor the network to determine the number of untrusted packets sent through malicious nodes. As can be seen from Table 5.8, the wormhole attack can transfer nearly 98.4% of the packets (for 10 nodes with a minimum number of hops are 6), 99.6% when the number of nodes is 32 and the minimum number of hops is 10. Because malicious nodes can change or drop packets, we must protect this network against wormhole attacks.

| <i>Time(seconds)</i> | <i>Generated</i> | <i>Trusted</i> | <i>Untrusted</i> |
|----------------------|------------------|----------------|------------------|
| 1 | 73 | 12 | 61 |
| 2 | 135 | 12 | 123 |
| 3 | 190 | 3 | 187 |
| 4 | 256 | 8 | 248 |
| 5 | 316 | 5 | 311 |
| 6 | 375 | 3 | 372 |
| 7 | 439 | 3 | 436 |
| 8 | 509 | 11 | 498 |
| 9 | 567 | 7 | 560 |
| 10 | 632 | 10 | 622 |

Table 5. 8 Effect of wormhole attack in AODV

5.5.1 Wormhole Detection Rate

The wormhole detection rate is defined as the ratio of detected wormholes to the total number of adversary attacks in the network. Figure 5.14 demonstrates the wormhole detection rate as a function of tunnel length. In each run, we randomly insert two wormholes into the network. This can be seen, our scheme (WDPT) is capable of detecting nearly 90% of wormhole attacks. The ANS method [42], on the other hand, detects wormholes at a rate of about 80%. When the tunnel length reaches 10, the RRS [42] method has a minimum rate of wormhole detection of about 50%. In general, the WDPT method performs satisfactorily and better than the other methods in detecting wormhole attacks. The detection accuracy of Reverse Routing Scheme (RRS) wormhole detection, Authentication of Nodes Scheme (ANS), and our proposed WDPT are shown in Fig 5.14.

$$DR = \text{Number of successful detection} / \text{Total number of attack} \text{ ----- (Equation 5.1)}$$

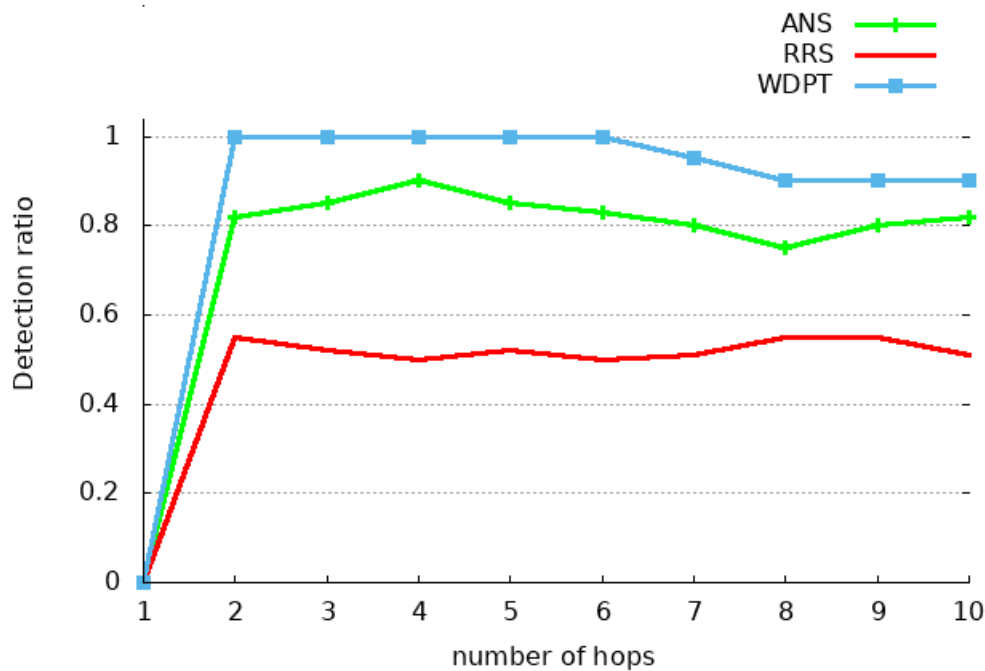


Fig 5. 7 Detection rate comparison at different hops

The detection accuracy of our proposed method is better than both ANS and RRS as we see from the above diagram. In this simulation, we launched 19 wormhole attacks for 10 nodes and our method detected and removed 17 of them. Which is nearly 89.47% of attacks. One of the major issues is false results (either false positives or false negatives).

⇒ False Positive (FP): number of normal instances that are normal but our method incorrectly predicted as an attack.

⇒ False Negative (FN): number of attack instances that are actual attacks but our method incorrectly predicted as normal.

Based on the above perspectives 10.526 % of our method was incorrectly predicted as a false negative and 5.26% false result of incorrectly classified normal nodes as wormhole nodes.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The rapid development of electronic devices and extensive dependence on Internet-based applications for both business and pleasure activities has led to an ever-increasing network of Internet communications. However, security issues have been one of the most critical problems for organizations as attackers also increased from time to time. It is spatially distributed hence it must have a low cost because those sensors have limited batteries, computational ability, and memory size. As a result, its limited ability to implement common security measures makes it vulnerable to a variety of attacks. There are various types of attacks that target various network layers. A wormhole attack, for example, is a dangerous and easily deployable attack that targets the routing layer. A wormhole attack is defined as generating an untrusted network shortcut. When two intruder sensors establish a wired or wireless connection, this is formed. This paper explained a transmission range-based and hop count method to detect this attack. The method is applied to the AODV protocol. The method has two stages that are applied for all sensors in the elected path of transmission. This method has an extremely high detection accuracy.

Wormhole attacks can be detected with help of the maximum transmission range of nodes and hop count method. High transmission power is a mode of wormhole attack. The existing methods of detection of wormhole attacks under this mode are not so cost-effective and are a little complex to understand. The proposed work is easy to understand and is cost-effective because transmission range and hop count methods are an inbuilt circuitry of a sensor node that can easily detect the malicious node by the number of hops during route request and route reply. The detection rate of our method is very high when the tunnel length is increased.

6.2 Contributions

✚ The contributions of this research work are:

- ⇒ Enhancement in the detection accuracy of previous works related to wormhole attack detection systems in wireless sensor networks.
- ⇒ Prevention of malicious nodes (wormhole nodes) to achieve the confidentiality of the network without degrading the performance compared with the previous work.

⇒ Introducing a new way to find the minimum number of hops from source to destination in AODV protocol to check the presence of a wormhole tunnel (since a wormhole tunnel is much longer than the normal route, the routing hop will be smaller than the normal route).

6.3 Future Works

Wormhole attacks can be performed in various modes as mentioned in this paper. In the future, this research can be protracted to detect wormhole attacks in those modes too. Sensor nodes have limited power. While implementing security, the power of sensors might be compromised. So, this research can also be extended in the direction of optimizing the power efficiency of the sensor nodes. In this paper, we only discuss a pair of malicious nodes that launch a wormhole attack. In the following work, we will study how to detect wormhole attacks when there is more than one wormhole link in the network topology. Moreover, we will study how to detect invisible wormhole attacks in the network.

Since wireless sensor nodes are ad-hoc in nature, the performance of the network will decrease from the normal mode while implementing a secured routing mechanism in wireless sensor networks, it is better to consider the performance of the network to alleviate throughput and other performance metrics.

REFERENCES

- [1]. Karlof, C. and Wagner, D., 2003. *Secure routing in wireless sensor networks: Attacks and countermeasures*. *Ad hoc networks*, 1(2-3), pp.293-315.
- [2]. Hu, Y.C., Perrig, A. and Johnson, D.B., 2003, March. *Packet leashes: a defense against wormhole attacks in wireless networks*. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428) (Vol. 3, pp. 1976-1986)*. IEEE.
- [3]. Khalil, I., Bagchi, S. and Shroff, N.B., 2005, June. *LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks*. In *2005 International Conference on Dependable Systems and Networks (DSN'05) (pp. 612-621)*. IEEE.
- [4]. kumar Dwivedi, R., Sharma, P. and Kumar, R., 2018, November. *A scheme for detection of high transmission power-based wormhole attack in WSN*. In *2018 5th IEEE Uttar Pradesh section international conference on electrical, electronics and computer engineering (UPCON) (pp. 1-6)*. IEEE.
- [5]. Ronghui, H., Guoqing, M., Chunlei, W. and Lan, F., 2009. *Detecting and locating wormhole attacks in wireless sensor networks using beacon nodes*. *International Journal of Computer and Information Engineering*, 3(7), pp.1741-1745.
- [6]. Chiu, H.S. and Lui, K.S., 2006, January. *DelPHI: wormhole detection mechanism for ad hoc wireless networks*. In *2006 1st international symposium on Wireless pervasive computing (pp. 6-pp)*. IEEE.
- [7]. Aldhobaiban, D., Elleithy, K. and Almazaydeh, L., 2014, November. *Prevention of wormhole attacks in wireless sensor networks*. In *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation (pp. 287-291)*. IEEE.
- [8]. Ghugar, U. and Pradhan, J., 2019. *A Review on Wormhole Attacks in Wireless Sensor Networks*. *International Journal of Information Communication Technology and Digital Convergence*, 4(1), pp.32-45.

- [9]. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., 2002. A survey on sensor networks. *IEEE Communications magazine*, 40(8), pp.102-114.
- [10]. Vieira, M.A.M., Coelho, C.N., da Silva, D.J. and da Mata, J.M., 2003, September. Survey on wireless sensor network devices. In *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 03TH8696) (Vol. 1, pp. 537-544). IEEE.*
- [11]. Tubaishat, M. and Madria, S., 2003. Sensor networks: an overview. *IEEE potentials*, 22(2), pp.20-23.
- [12]. Choi, S., Kim, D.Y., Lee, D.H. and Jung, J.I., 2008, June. WAP: Wormhole attack prevention algorithm in mobile ad hoc networks. In *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (suc 2008) (pp. 343-348). IEEE.*
- [13]. Kaffashi, E., Mousavi, A., Rahvard, H., Bojnordi, S.H., Khademsadegh, F. and Amirian, S., 2015. A new attack on link-state database in open shortest path first routing protocol. *Journal of Electrical and Electronic Engineering*, 3(2-1), pp.39-45.
- [14]. Hu, L. and Evans, D., 2004, February. Using directional antennas to prevent wormhole attacks. In *NDSS (Vol. 4, pp. 241-245).*
- [15]. Lazos, L. and Poovendran, R., 2004, October. SeRLoc: Secure range-independent localization for wireless sensor networks. In *Proceedings of the 3rd ACM Workshop on Wireless Security (pp. 21-30).*
- [16]. Wang, W. and Bhargava, B., 2004, October. Visualization of wormholes in sensor networks. In *Proceedings of the 3rd ACM workshop on Wireless security (pp. 51-60).*
- [17]. WGRP, L., 2007. A lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks, Issa Khalil, Saurabh Bagchi and Ness B.
- [18]. Awerbuch, B., Curtmola, R., Holmer, D., Rubens, H. and Nita-Rotaru, C., 2005, September. On the survivability of routing protocols in ad hoc wireless networks. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05) (pp. 327-338). IEEE.*

- [19]. Song, N., Qian, L. and Li, X., 2005, April. *Wormhole attacks detection in wireless ad hoc networks: A statistical analysis approach*. In *19th IEEE international parallel and distributed processing symposium* (pp. 8-pp). IEEE.
- [20]. Gorlatova, M.A., Mason, P.C., Wang, M., Lamont, L. and Liscano, R., 2006, October. *Detecting wormhole attacks in mobile ad hoc networks through protocol breaking and packet timing analysis*. In *MILCOM 2006-2006 IEEE Military Communications Conference* (pp. 1-7). IEEE.
- [21]. Rasmussen, K.B. and Capkun, S., 2007, September. *Implications of radio fingerprinting on the security of sensor networks*. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007* (pp. 331-340). IEEE.
- [22]. Win, K.S., 2008. *Analysis of detecting wormhole attack in wireless networks*. In *World Academy of Science, Engineering and Technology*.
- [23]. Nait-Abdesselam, F., Bensaou, B. and Taleb, T., 2008. *Detecting and avoiding wormhole attacks in wireless ad hoc networks*. *IEEE Communications Magazine*, 46(4), pp.127-133.
- [24]. Özdemir, S., Meghdadi, M. and Güler, Ý., 2008. *A time and trust-based wormhole detection algorithm for wireless sensor networks*. In *3rd Information Security and Cryptology Conference (ISC'08)* (pp. 139-142).
- [25]. Krontiris, I., Giannetsos, T. and Dimitriou, T., 2008, September. *LIDeA: a distributed lightweight intrusion detection architecture for sensor networks*. In *Proceedings of the 4th international conference on Security and privacy in communication networks* (pp. 1-10).
- [26]. Choi, S., Kim, D.Y., Lee, D.H. and Jung, J.I., 2008, June. *WAP: Wormhole attack prevention algorithm in mobile ad hoc networks*. In *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)* (pp. 343-348). IEEE.
- [27]. Özdemir, S., Meghdadi, M. and Güler, Ý., 2008. *A time and trust based wormhole detection algorithm for wireless sensor networks*. In *3rd Information Security and Cryptology Conference (ISC'08)* (pp. 139-142).

- [28]. Vu, H., Kulkarni, A., Sarac, K. and Mittal, N., 2008, October. *Wormeros: A new framework for defending against wormhole attacks on wireless ad hoc networks*. In *International Conference on Wireless Algorithms, Systems, and Applications* (pp. 491-502). Springer, Berlin, Heidelberg.
- [29]. Sankaran, M.S., Poddar, S., Das, P.S. and Selvakumar, S., 2009. *A novel security model saw: Security against wormhole attack in wireless sensor networks*. In *Proceedings of International Conference on PDCN*.
- [30]. Triki, B., Rekhis, S. and Boudriga, N., 2009, July. *Digital investigation of wormhole attacks in wireless sensor networks*. In *2009 Eighth IEEE International Symposium on Network Computing and Applications* (pp. 179-186). IEEE.
- [31]. Chen, H., Lou, W. and Wang, Z., 2009, July. *Conflicting-set-based wormhole attack resistant localization in wireless sensor networks*. In *International Conference on Ubiquitous Intelligence and Computing* (pp. 296-309). Springer, Berlin, Heidelberg.
- [32]. Roy, D.B., Chaki, R. and Chaki, N., 2010. *A new cluster-based wormhole intrusion detection algorithm for mobile ad-hoc networks*. *arXiv preprint arXiv:1004.0587..*
- [33]. Chen, H., Lou, W., Sun, X. and Wang, Z., 2009. *A secure localization approach against wormhole attacks using distance consistency*. *EURASIP Journal on Wireless Communications and Networking*, 2010, pp.1-11.
- [34]. Prasannajit, B., Anupama, S., Vindhykumari, K., Subhashini, S.R. and Vinitha, G., 2010, August. *An approach towards detection of wormhole attack in sensor networks*. In *2010 First International Conference on Integrated Intelligent Computing* (pp. 283-289). IEEE.
- [35]. Rehmani, M.H. and Saleem, Y., 2015. *Network simulator NS-2*. In *Encyclopedia of Information Science and Technology, Third Edition* (pp. 6249-6258). IGI Global.
- [36]. Lee, G., Seo, J. and Kim, D.K., 2008, April. *An approach to mitigate wormhole attack in wireless ad hoc networks*. In *2008 international conference on information security and assurance (ISA 2008)* (pp. 220-225). IEEE.
- [37]. Win, K.S., 2008. *Analysis of detecting wormhole attack in wireless networks*. In *World Academy of Science, Engineering and Technology*.

- [38]. Buratti, C. and Verdone, R., 2013, April. P-CSMA: A priority-based CSMA protocol for multi-hop linear wireless networks. In *European Wireless 2013; 19th European Wireless Conference* (pp. 1-8). VDE.
- [39]. *Ethiopian Grade 11 mathematics*, 2002, Federal Democratic Republic of Ethiopia Ministry of Education, Ch-3, coordinate geometry 69-111.
- [40]. Wang, Y., Lin, J., Annavaram, M., Jacobson, Q.A., Hong, J., Krishnamachari, B. and Sadeh, N., 2009, June. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services* (pp. 179-192).
- [41]. Bareth, U. and Kupper, A., 2011, July. Energy-efficient position tracking in proactive location-based services for smartphone environments. In *2011 IEEE 35th Annual Computer Software and Applications Conference* (pp. 516-521). IEEE.
- [42]. Poornima, E. and Bindhu, C., 2010. Prevention of Wormhole Attacks in Geographic Routing Protocol. *International Journal of Computer Network and Security (IJCNS)*, 3(1), pp.42-50.
- [43]. <http://www.google.com>. (2022), how long does a GPS signal take, USA ocean and satellite study, <http://www.socreatic.org>
- [44]. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., 2002. Wireless sensor networks: a survey. *Computer networks*, 38(4), pp.393-422.
- [45]. Chong, C.Y. and Kumar, S.P., 2003. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8), pp.1247-1256.
- [46]. Rault, T., Bouabdallah, A. and Challal, Y., 2014. Energy efficiency in wireless sensor networks: A top-down survey. *Computer networks*, 67, pp.104-122.
- [47]. Perkins, C., Belding-Royer, E. and Das, S., 2003. Ad hoc on-demand distance vector (AODV) routing (No. rfc3561).
- [48]. Buttyán, L., Dóra, L. and Vajda, I., 2005, July. Statistical wormhole detection in sensor networks. In *European Workshop on Security in Ad-hoc and Sensor Networks* (pp. 128-141). Springer, Berlin, Heidelberg.

Appendix

Sample of TCL file

To launch a wormhole attack inside tcl file

```
[$n4 set ll_(0)] wormhole-peer [$n6 set ll_(0)]  
[$n6 set ll_(0)] wormhole-peer [$n4 set ll_(0)]
```

To Calculate distance between nodes

```
for {set i 0} {$i < $val(nn)} {incr i} {  
for {set j 0} {$j < $val(nn)} {incr j} {  
set dx [expr $xx($i)-$xx($j)]  
set dy [expr $yy($i)-$yy($j)]  
set dx2 [expr $dx * $dx]  
set dy2 [expr $dy * $dy]  
set h2 [expr $dx2 + $dy2]  
set h($i-$j) [expr (pow($h2, 0.5))]  
set nhop [expr ceil($h($i-$j)/$max_range)]  
puts "distance of node($i) from node($j) = $h($i-$j)"  
puts $range "$i\t$j\t$h($i-$j)\t$nhop"}}
```

Sample of C++ program

To create wormhole attack in MAC

```
ch->addr_type() = NS_AF_ILINK;  
ch->next_hop() = MAC_BROADCAST;  
ll->sendDown( p );  
hdr_cmn::access(llinfo->hold_)->addr_type() = NS_AF_ILINK;  
hdr_cmn::access(llinfo->hold_)->next_hop() = ah->arp_sha;  
ll->sendDown( llinfo->hold_ );
```

```
llinfo->hold_ = 0;
ch->addr_type() = NS_AF_ILINK;
ch->next_hop() = ah->arp_sha;
ll->sendDown( p );
```

To store, check run time information and prevent in AODV protocol

```
xpos=ypos=zpos=0.0;
n_time=0.0;
energy_t=0.0;
n_speed=0.0;
attacker=false;
fpt=fopen("runtimetracer.dat","wr");
fpt1=fopen("activenodes.dat","wr");
fprt=fopen("route_table.dat", "wr");
ss.open("runtimetracer.dat");
t_count=0;
tfile.open("route_table.dat");
row=0,arow=0,worm1=0,worm2=0;
t_node=(MobileNode*)(Node::get_node_by_address(index));
((MobileNode*) t_node)->getLoc(&xpos,&ypos,&zpos);
n_time=((MobileNode*)t_node)->getUpdateTime();
//fprintf(fpt, "\t%d\t%d\t%f\t%d\t%d\t%d\n",int(n_time),index,energy_t,int(xpos),int(ypos)
,n_speed);
fprintf(fpt, "%f\t%d\t%f\n",n_time,index,energy_t);
t_node=(MobileNode*)(Node::get_node_by_address(index));
n_speed=((MobileNode*)t_node)->speed();
energy_t=t_node->energy_model()->energy();
fprintf(fpt1, "%d\n",index);
```

```
anodes[arow]=index;
arow++;
ifstream ifs("worm.txt");
if(ifs.is_open()){
ifs >> worm1 >> worm2;
}else{
cout<<"Unable to open wormlist file"<<endl;
}

if(worm1!=0 && worm2!=0){
if(worm2==9999){
cout<<"The Wormhole is: "<<worm1<<endl;
if(index==worm1){
drop(p,DROP_RTR_NO_ROUTE);
}
}else{
cout<<"The first Wormhole is: "<<worm1<<endl;
cout<<"The second Wormhole is: "<<worm2<<endl;
if(index==worm1 ||index==worm2 ){
drop(p,DROP_RTR_NO_ROUTE);
}}
void AODV::rt_print(nsaddr_t nodeid) {
aadv_rt_entry *rt;
for (rt=rtable.head();rt; rt = rt->rt_link.le_next) {
fprintf(fprrt,"%i\t%f\t%i\t%i\t%i\t%i\t%f\n", nodeid, CURRENT_TIME, rt->rt_dst, rt->rt_nexthop, rt->rt_hops, rt->rt_seqno, rt->rt_expire);
}
}
```

To detect wormhole attack

```
bool sortcol(const vector<int>& v1, const vector<int>& v2)
{
    return v1[2] < v2[2];
}

bool esortcol(const vector<float>& v1, const vector<float>& v2)
{
    return v1[1]
}

std::vector<std::vector<float>> > allData;
std::vector<std::vector<float>> > rtable;
std::vector<std::vector<float>> > dist;
std::vector<std::vector<int>> > dcheck;
std::vector<std::vector<float>> > echeck;
float dsource,nod[100][2];
bool worm=false;

//check if wormhole attack exist based on number of hops
for(int i=0;i<rtable.size();i++){
for(int j=0;j<dist.size();j++){
    if(rtable[i][0]==dist[j][0] && rtable[i][2]==dist[j][1]){
        if(rtable[i][4]<dist[j][3]){
            worm=true;
            // worms[wt]=rtable[i][3];
            //wt++;
            //dsource=dist[j][2];
            //ss=rtable[i][0];
```

```
        //dd=rtable[i][2];

    }

}

}

}

if(worm==false){

    cout<<"There is no wormhole attack"<<endl;

    }if(wnop==2){

    worml=nhop;

    if(wlist.is_open()){

wlist<<worml<<"\t"<<9999<<endl;

    }else{

    cout<<"Unable to open worm list file"<<endl;

    }

    cout<<"*\tThe wormhole is: "<<worml<<"\t"<<"\n*\tand There is no second wormhole found in the network"<<endl;

}

else

    if(wnop==3){

    worml=nhop;

    int temp;

    for(int i=0;i<allData.size();i++){

        for(int j=0;j<allData[i].size();i++){

            for(int k=i+1;k<allData.size();k++){

                if(allData[i][2]>=allData[k][2]){
```

```
temp=allData[i][j];
allData[i][j]=allData[k][j];
allData[k][j]=temp;
worm2=allData[i][1];
    }
  }
}
}
if(worm1==worm2){
for(int i=0;i<dcheck.size();i++){
worm2=dcheck[i][1];
}
}
```