TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: 074MSICE014

PROVENANCE BASED MALICIOUS NODE DETECTION IN WIRELESS SENSOR NETWORK USING BLOOM FILTER

by

Roshan Kandel

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

AUGUST, 2021

“Provenance Based Malicious Node Detection in Wireless Sensor Network Using Bloom Filter”

by

Roshan Kandel

074/MSICE/014

Thesis Supervisor

Asst. Prof. Daya Sagar Baral

A final thesis report submitted in partial fulfillment of the requirements for the degree of Master of Science in Information and Communication Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

August, 2021

COPYRIGHT

DECLARATION

I declare that the work hereby submitted for Master of Science in Information and Communication Engineering (MSICE) at IOE, Pulchowk Campus entitled **"Provenance Based Malicious Node Detection in Wireless Sensor Network Using Bloom Filter"** is my own work and has not been previously submitted by me at any university for any academic award. I authorize IOE, Pulchowk Campus to lend this thesis to other institution or individuals for the purpose of scholarly research.

Roshan Kandel
074/MSICE/014
Date: August, 2021

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled **"Provenance Based Malicious Node Detection in Wireless Sensor Network Using Bloom Filter"** submitted by **Roshan Kandel** in partial fulfillment of the requirement for the award of the degree of "Master of Science in Information and Communication Engineering".

.....

Supervisor: Daya Sagar Baral

Assistant Professor

Department of Electronics and Computer Engineering

.....

External Examiner: Er. Kumar Pudasainee

Chief Technical Officer (CTO)

Agriculture Development Bank

.....

Committee Chairperson: Dr. Basanta Joshi

Assistant Professor

Department of Electronics and Computer Engineering,

Date of Approval: August, 2020

DEPARTMENTAL ACCEPTANCE

The thesis entitled **"Provenance Based Malicious Node Detection in Wireless Sensor Network Using Bloom Filter"**, submitted by **Roshan Kandel** in partial fulfillment of the requirement for the award of the degree of **"Master of Science in Information and Communication Engineering"** has been accepted as a bonafide record of work independently carried out by him in the department.

.....

Prof. Dr. Ram Krishna Maharjan
Head of the Department
Department of Electronics and Computer Engineering
Pulchowk campus, Tribhuvan University, Nepal

ACKNOWLEDGEMENT

I owe a great deal to express thanks to Asst. Prof. **Daya Sagar Baral** Sir for his continuous support and guidance which has provided me constant motivation and confidence to carry out my work.

I am very grateful to Prof. Dr. Ram Krishna Maharjan Sir, Head of Department of Electronics and Computer Engineering, Pulchowk Campus for his precious support.

I would like to thank Dr. Basanta Joshi Sir, Coordinator of MSc Information and Communication Engineering, Pulchowk Campus, for his constant effort on thesis and course related activities.

I am also very grateful to Prof. Dr. Shashidhar Ram Joshi Sir, Prof. Dr. Subarna Shakya Sir, Dr. Dibakar Raj Pant Sir**,** Assoc. Prof. Dr. Surendra Shrestha, Assoc. Prof. Dr. Sanjeeb Prasad Panday Sir, Assoc. Prof. Dr. Nanda Bikram Adhikari Sir, Dr. Aman Shakya Sir, Dr. Arun Timalsina Sir, Sharad Kumar Ghimire Sir, Babu Ram Dawadi Sir and other faculties for their valuable suggestions, support and technical expertise.

Roshan Kandel
074-MSICE-014

ABSTRACT

Wireless Sensor Networks (WSN) are essential in modern day to gain information about the environmental studies that are useful in decision making. Data is transferred through wireless medium, so the integrity of the data has to be maintained. We have created a model that helps to securely transmit the data from source node to base station. We have implied provenance based bloom filter and cryptographic algorithm to securely transmit data to the base station. We have also created a model to detect packet dropping malicious node when data is in transit. In our model, we have used AODV protocol as a routing algorithm and AES-128 as a cryptographic algorithm to encrypt the data during transmission. Bloom filter requires a hashing algorithm so in our case SHA-224 cryptographic hashing algorithm has been deployed.

We used a light weight Bloom filter model to transmit the data and to detect any packet dropping node which act as an intermediate nodes. We used provenance information to help detect any malicious packet dropping nodes. We relied on provenance encoding and decoding methods to our model. We analyzed our model using different parameters like Verification failure rate (VFR), True false positive (TFP), throughput, end to end delay, etc on different number of nodes, packet size and bloom filter size.

Key words: WSN, Malicious Node, Bloom Filter, Provenance, AODV, AES, SHA, Packet Drop

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

WSN: Wireless Sensor Network

AODV: Ad hoc On-Demand Distance Vector

AES: Advanced Encryption Standard

SHA: Secure Hash Algorithm

NS: Network Simulator

BF: Bloom Filter

VPN: Virtual Private Network

NSA: National Security Agency

CPU: Central Processing Unit

ADC: Analog to Digital Converter

MD: Message Digest

URL: Uniform Resource Locator

CDN: Content Delivery Network

RREQ: Route Request

RREP: Route Reply

MAC: Message Authentication Code

VID: Vertex Identifier

BS: Base Station

UDP: User Datagram Protocol

VFR: Verification Failure Rate

TFP: True False Positive

PDR: Packet Delivery Ratio

SDN: Software Defined Network

IoT: Internet of Things

1. INTRODUCTION

1.1 Background

Sensor networks are used in numerous applications like environmental condition monitoring, medical, military surveillance etc. The data captured in sensor networks are to be sent to the base station for analysis of data. During transmission of data, we need to pass these data through wireless medium to the base station followed by many intermediate sensor networks called intermediate nodes. So, wireless sensor network (WSN) is the mechanism used for transferring data from sensor device i.e. source node to base station i.e. destination node using wireless communication. Sensor nodes can be deployed in extreme environmental condition where human involvement is difficult. So, the main concern in WSN is the security of data while transferring from source node to destination. Intruders may attack the node and can get easy access to the data.

Malicious nodes are the affected nodes which shows malicious behavior. Nodes can be subjected to both physical damage and logical damage. Logical damage in a sense that it may be attacked by intruder or nodes itself can be selfish so that they don't transmit the received data from its neighboring nodes to the succeeding nodes. So the data need to be routed through alternative best route with minimum path to the destination node. In this work, we are going to discuss about the best route to the destination node whenever there is a malicious node present using provenance based bloom filter.

Data generated at source sensor node needs to be securely transferred to the base station so that correct decision making can be done. The diversity of data source make the fact that trustworthiness of data is maintained. Recent research have proven that provenance based mechanism have played important role in WSN. Low energy, efficient storage, bandwidth consumption and secure transferring of data are several challenges that needs to be addressed by provenance based system. Because it investigates a past and extensive history of the ownership of data and activities conducted on data, data provenance is an adequate technique for determining the trustworthiness of data. Provenance ensures data integrity, which is essential for making accurate decisions in a variety of crucial applications such as military

applications. In a multi-hop sensor network, data provenance allows the base station to track the source and forwarding path of an individual data packet since its origination. The restricted storage, energy, and bandwidth limits of the sensor nodes present significant hurdles in recording provenance for each data packet. As a result, a light-weight provenance solution that does not impose considerable overhead is required. It's also critical to consider security concerns including confidentiality, integrity, and provenance freshness. Our primary goal is to create a provenance encoding and decoding technology that is secure. We deployed a provenance encoding method in which each node along a data packet's journey securely embeds provenance information in a Bloom filter, which is then transmitted with the data. The base station extracts and verifies the provenance information after receiving the packet.

Our system uses Bloom Filter (BF) in coordination with provenance information to securely transmit data from sensor node to base station. A Bloom filter is **a** space-efficient probabilistic data structure that is used to test whether an element is a member of a set [1]. For example, checking availability of username is set membership problem, where the set is the list of all registered username. BF generally discusses about the presence and absence of searched data in its set. If searched element is present in set then it can't assure the presence of that element i.e. both True positive and False positive can take place. But when we check unavailability of data then it as assure about the fact 100%. So there is a case for True negative only. In our model, we have used this filter so assure that the element is in set or not. In our model, we have used Bloom filters (BF), which are fixed-size data structures that compactly represent provenance. Bloom filters make efficient usage of bandwidth, and they yield low error rates in practice.

At first, empty bloom filter is used which bit array of m bits, all set to zero, like this –

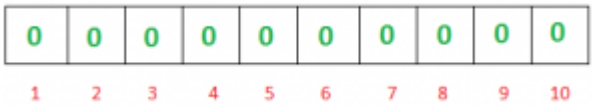

Figure 1: Bloom Filter

We need “k” number of hash functions to calculate the hashes for a given input. When we want to add an item in the filter, the bits at k indices h1(x), h2(x), … hk(x) are set, where indices are calculated using hash functions.

Advanced Encryption Standard (AES) is a type of cipher that is used to protect data in communication. It is one of the best encryption algorithm which combines speed and security. Nowadays, we find AES encryption in many applications that we use today. AES is a symmetric encryption where two parties use same key to encrypt and decrypt that message. It uses multiple rounds to encrypt the data. So, this is why it is hard to break. There are 3 lengths of AES encryption keys i.e. 128-bit key, 192-bit key and 256-bit key. Despite of the different key length, block size is 128 bit long. In our model we have used AES-128 bit key to encrypt the data in communication.

There are various encryption standards like DES, etc. We have used AES because it requires less memory to operate. AES is used in our wi-fi that we use, VPN, mobile applications, compression applications, password managers and many more applications that we use today. AES decryption is an inversion of AES encryption that restores cipher text to plain text. As, we know that AES uses symmetric key, same key is used for encryption and decryption.

The Secure Hash Algorithm (SHA) is a collection of cryptographic functions designed to keep data safe. It operates by converting data using hash functions, which result in a fixed-length string. These methods are one-way functions, which means that once they've been turned into hash values, they're nearly impossible to reverse.

Some of SHA algorithms are SHA -1, SHA-2 and SHA-3 whereas SHA-0 is obsolete. SHA is widely used to encrypt passwords. The encrypted passwords are stored in the form of passwords not the actual password. Even if the attacker compromises the database, they will get only hash values. Additionally, SHA shows avalanche effect, where changing few letters that are encrypted causes a huge change in output. So, due to this effect, hash values do not give any information regarding input string.

In the field of cryptography, SHA-1 algorithm is a crypto formatted hash function that produces a string of 160 bits i.e. 20 byte hash value over any input length.
The hash valued thus produced is known as message digest. These hash functions are used to maintain and save data in a secure manner by offering three types of characteristics: pre-image resistance, also known as first level image resistance, second level image resistance, and collision resistance. The pre-image approach makes it difficult and time-consuming for the hacker to locate the original message using hash values.
Second resistance technique is used so that it is difficult for the attacker to go through decoding the error message that are produced during decryption when first level is decrypted. And the last stage is collision resistance, which makes difficult for the attacker to find two completely different messages which hash to the same value.

As, we have already discussed the types of SHA. SHA-1 is a 160 bit long or 20 byte long hashed function to digest the message. It was designed and developed by National Security Agency (NSA). Weakness to this cryptographic technique is found and was replaced by SHA-2.
Due to the exposed vulnerabilities of SHA-1, researchers modified the algorithm to produce SHA-2, which consists of not one but two hash functions known as SHA-256 and SHA-512, using 32- and 64-bit words, respectively. There are additional truncated versions of these hash functions, known as SHA-224, SHA-384, SHA-512/224, and SHA-512/256, which can be used for either part of the algorithm. SHA-1 and SHA-2 differ in several ways; mainly, SHA-2 produces 224- or 256-sized digests, whereas SHA-1 produces a 160-bit digest; SHA-2 can also have block sizes that contain 1024 bits, or 512 bits, like SHA-1. In our model, we have used SHA -224 hashing with bloom filter to protect the provenance information in bloom filter as suggested by [2]. Experiment done by the author shows that SHA -224 produces best output and in quick execution time.

1.2 Problem Statement

Transferring data from source node to destination node should address several security issue so that the data received at the base station should have freshness, integrity and

confidentiality. Many researches have been done to counter security issues using various mechanisms and algorithms for single malicious node detection. But few of the research have been done in case where we have multiple malicious nodes. Different researchers and scholars have appended various challenges such as memory usage, usages of bandwidth, and attack by the malicious data, maintaining the originality and the integrity of the data.

Sensor nodes can be deployed in extreme environmental condition where human involvement is difficult. So, the main concern in WSN is the security of data while transferring from source node to destination. Intruders may attack the node and can get easy access to the data. In researches done till date, few of the mechanisms exists where packet is re-routed using best alternative shortest path.

1.3 Objective

The main objectives of this thesis are as follows:

- To detect multiple malicious node in WSN when message is sent from source node to base station.
- To securely transmit data from source node to destination node.

1.4 Organization of the thesis

The thesis is organized as mentioned below: •

- Chapter 2: provides information about WSN, Bloom Filter, AODV & other key terminologies.
- Chapter 3: gives information regarding previous work related to WSN malicious node detection.
- Chapter 4: The methodology used for detecting malicious node is discussed in the chapter.
- Chapter 5: The results obtained using NS3 simulator are discussed as well as analyzed in this chapter.
- Chapter 6: The last chapter draw conclusion from the analysis of the results and suggest some recommendations.

2. WSN, BLOOM FILTER, AODV & KEY TERMINOLOGIES.

2.1 Wireless Sensor Networks

Wireless Sensor Network (WSN) is a standard service deployed in commercial and industrial applications. It is made up of nodes that detect and store the environment, such as temperature, pressure, sound, humidity, and so on. These data can be employed in real-time applications for tasks such as smart detection, neighbor node discovery, data processing and storage, data collecting, target tracking, monitor and control, synchronization and effective routing between the base station and nodes. These technologies can be used in numerous applications like medical, environmental, military, crisis management, etc.

WSN is a type of wireless network that has large number of minute, low powered electronic devices called nodes. These nodes are used to capture large amount of node using sensors inside it and these data can be effective in monitoring the activities where sensors are deployed and helps to take necessary decisions. Nodes can be also called as tiny computers that are joint to form a network. The sensor node is multifunctional and energy efficient device. Each node sense the data and can be also called as transceiver which can also receive and forward the data. In other words, these sensor nodes can act as a router. Generally, sensor nodes in WSN are fixed and there are some applications where there are mobile nodes too. In contrast to ad hoc network where there are fewer nodes, WSN has large number of nodes in the range of hundreds to thousands.

2.1.1 Structure of a wireless sensor network

Structure of WSN includes different topologies that are deployed to create a network and hence transfer data between different nodes using radio waves. Different topologies of WSN are discussed below:

2.1.1.1 Star Network

A star network topology is a communication topology in which a single base station can send and/or receive a message to a number of remote nodes. The remote nodes can’t send

the message to each other. This type of topology has the advantage of low power consumption as all the nodes do not have to involve in each and every packet transmission. Limitation of this network is that base station has be within the communication range of all individual nodes and is not as robust as other topologies due to its dependency on a single node i.e. base station to manage the network.

2.1.1.2 Mesh Network

This sort of network allows data packets to be sent from one node in a network to another node within its communication range. This enables for multi-hop communication, which means that if a node wishes to deliver a message to another node that is out of radio transmission range, it can still use intermediate nodes to interact. Redundancy and extensibility are two advantages of these networks. If a single node fails then, there is still another path to the desired node using alternate route. In addition, the range of network is not limited by the range in between individual nodes, it can be easily expanded by adding more nodes to the network. The limitation of these networks are power consumption and end-to-end delay. As nodes have to participate in transferring data packet between the nodes that act as an intermediate node, they have to regularly consume some energy which cause to the decay of battery life. Battery life of WSN is usually low. Also, talking about end to end delay, the number of communicating nodes to the destination increases, time to deliver message also increases and hence delay is increased.

2.1.1.3 Hybrid star – Mesh network

This type of network combines star and mesh topologies which makes network robust and versatile while maintaining power consumption of nodes to minimum. The sensor nodes with the lowest power are not given the capacity to forward messages in this network design. This ensures that power usage is kept to a minimum. Other network nodes, on the other hand, have multi-hop capabilities, allowing them to relay messages from low-power nodes to other network nodes. Multi-hop nodes are often higher-power nodes that are often linked into the

electrical mains line if possible. This is the topology implemented by the up and coming mesh networking standard known as ZigBee [3].

2.1.2 Structure of a wireless sensor node

The four essential components of a sensor node are the sensing unit, processing unit, transceiver unit, and power unit. A sensing device detects the physical environment and sends the information to a central processing unit (CPU), which processes and stores the information. Sensors and analog to digital converters are included (ADC). Physical phenomena are turned into electrical signals, which are then translated to digital using an ADC. A processing unit is linked to a small storage unit and can control the operations that allow sensor nodes to work together to execute sensing activities. Processing unit consists of micro controller which performs execution on the data fed by sensor unit, execution of communication protocols, cryptographic tasks and sensor controlling tasks [4]. Transceiver unit connects node to the network. It consists of antenna i.e. transceiver to communicate with other nodes in a network. It also transmits the received data, processed data to other nodes so that data reaches base station. Power unit supplies electrical energy to the components of sensor node. Power unit generally consists of solar power cells or batteries which have lower

power storage capacity. Architecture of sensor node is shown in figure 2.

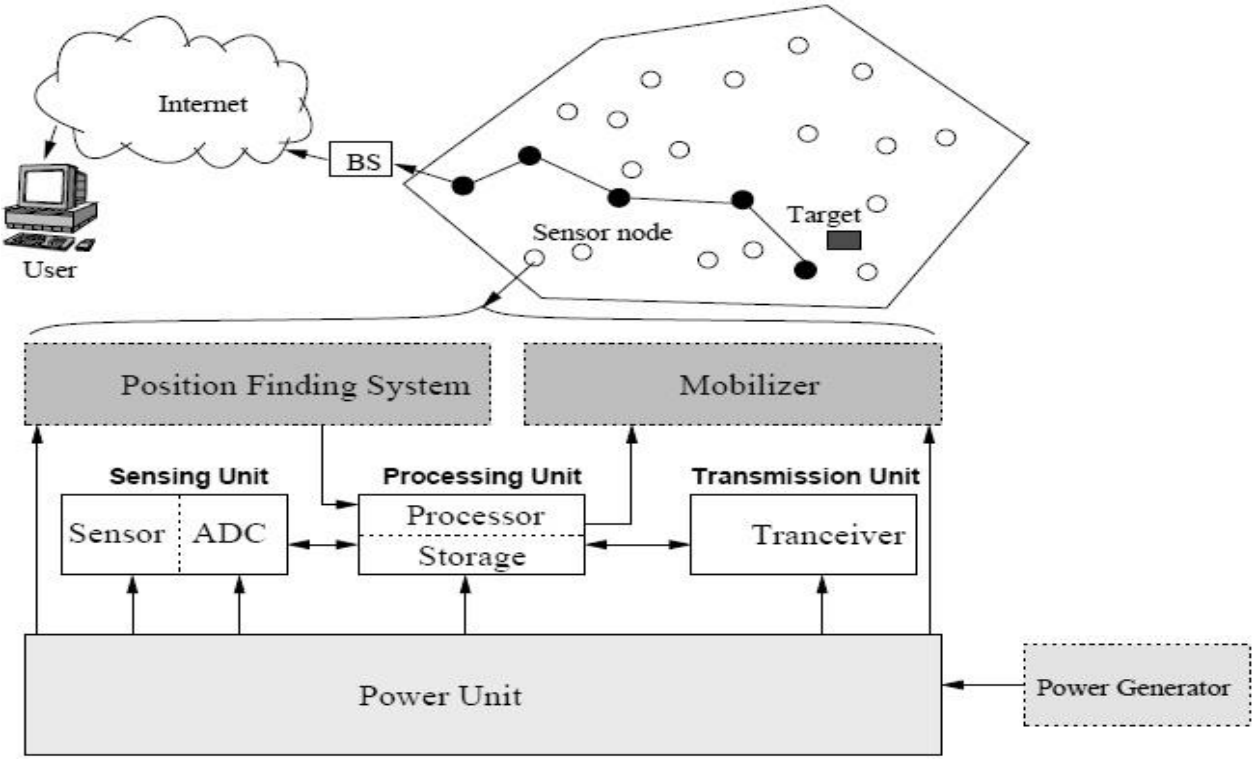

Figure 2: Structure of sensor node

2.1.3 Challenges of WSN

Sensor networks, which are a subset of wireless ad hoc networks, have various challenges in their adoption. Sensor nodes communicate across wireless, lossy lines in the absence of any infrastructure. Another difficulty is the sensor nodes' limited energy supply, which is typically non-renewable. The protocols must be designed considering energy in order to maximize lifetime.

Fault Tolerance: Sensor nodes are susceptible and regularly deployed in risky environments, requiring fault tolerance. Hardware problems, physical damage, or a lack of energy can all cause nodes to fail. In wired or infrastructure-based wireless networks, we expect a far higher rate of node failures than is commonly expected. The protocols of a sensor network should be able to detect these errors as fast as feasible and be strong enough to handle a significant number of failures while

keeping the network operational. This is especially critical in routing systems, which must ensure that alternate channels for packet rerouting are available.

Hardware Constraints: Every sensor node must have a sensing unit, a processing unit, a transmission unit, and a power supply, at the very least. To enable location-aware routing, the nodes may incorporate various built-in sensors or external devices, such as a localization system. Each new functionality, on the other hand, comes at a cost, with the node's power consumption and physical size growing. As a result, extra functionality must be regularly weighed against cost and energy use.

Sensor Network Topology: Despite the fact that WSNs have progressed in many aspects, they still have limited energy, processing capacity, memory, and communications resources. The most critical of these limits is energy consumption, as seen by the vast array of algorithms, techniques, and protocols developed to save energy and hence extend the network's lifetime. Topology maintenance is one of the most important difficulties being tackled to reduce energy consumption in wireless sensor networks.

Power Consumption: Many of the issues with sensor networks, as we've seen, relate around restricted power resources. The battery's size is limited by the size of the nodes. The concerns of efficient energy utilization must be properly considered in the software and hardware architecture. Data compression, for example, may save energy during radio transmission but consumes more energy during processing and/or filtering. The application also affects the energy strategy; in some cases, it may be appropriate to turn off a subset of nodes to save energy, whilst in other cases, all nodes must be operational at the same time.

Security: One of the difficulties in WSNs is meeting high security requirements while working with limited resources. A large number of wireless sensor networks collect sensitive data. Sensor nodes that are operated remotely and unmanaged are more vulnerable to hostile invasions and attacks. Node authentication and data

secrecy are two security criteria in WSNs. In order to detect both legitimate and non-legitimate nodes from a security aspect, the deployment sensors must pass a node authentication phase by their respective management nodes or cluster heads, and unauthorized nodes can be isolated from WSNs during the node authentication process. As a result, sensor networks will require novel solutions for key creation and distribution, node authentication, and secrecy.

2.1.4 Wireless Sensor Network Applications

Because of their adaptability in solving problems across a wide range of application areas, wireless sensor networks have increased in popularity, and they have the potential to change our lives in a variety of ways. WSNs have proven to be useful in a wide range of applications. Seismic, magnetic, thermal, optical, infrared, radar, and acoustic sensors are just a few of the numerous types of sensors that can be employed in wireless sensor networks to monitor a variety of environmental variables. Continuous sensing, event identification, event detection, and local actuator control are all done with sensor nodes. Wireless sensor networks are most commonly used in health, military, environmental, home, and other commercial applications.

2.2 Bloom Filter and its components

Let's say we want to determine if a particular element is present in a set of items. There are a variety of search algorithms that may be used to accomplish this. Even with an effective search method, storage becomes an issue when dealing with a large set (millions of elements). Due to disk access, there is also latency. One option is to use a bloom filter. Bloom filter is a data structure that stores the original set in a more compact format while also allowing set membership queries, i.e. determining whether an element is a member of the set. Bloom filter is a probabilistic data structure that saves space. With the rise in data since 2000, there is increase in interest to use Bloom Filter in different applications [5].

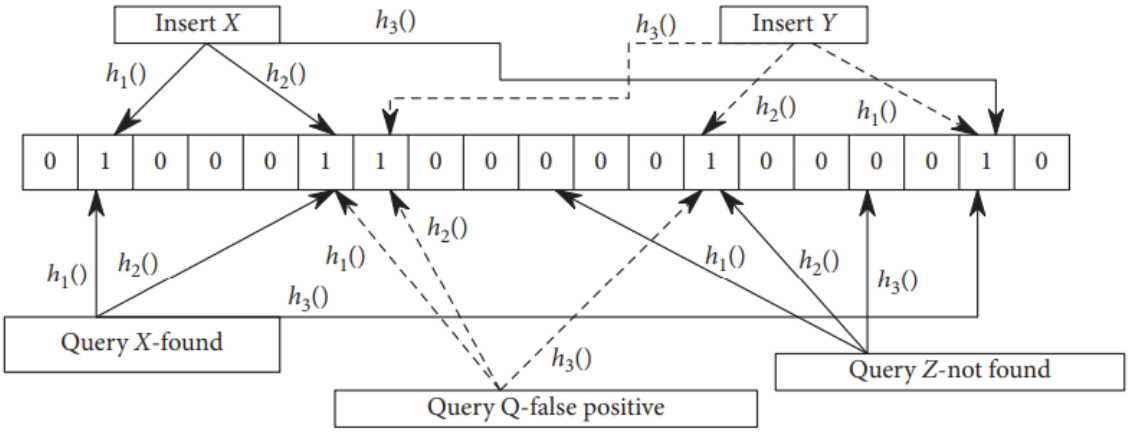

Figure 3: Bloom filter with 3 hash functions that illustrates the true positive, false positive, and true negative *[5]*.

A Bloom filter is simply a **bit array** of length m for storing elements of set S={x1,x2,…,xn}. The set is empty because the filter starts with all zeros. When you add a new element, it's hashed with c different hash functions. Each function will generate a bit location in the filter that will be set to one. As a result, each element will set the filter's **a** bits [1].

The requested element is passed through the **c** - hash functions when a query is conducted. The bit positions that result are examined in the filter. If at least one of these is zero, we know the element does not belong in the set. If all of the bits are ones, it's possible that it's part of the set. The filter is thus probabilistic. Because a query may verify against bit locations specified by more than one stored element, it is probabilistic. In figure 3, we can see that there are 3 hash function used with 19-bit bloom filter space size and element X & Y are inserted using c hash functions. When we query elements then the result is shown in figure 3. Element X and Q is found whereas X is True Positive, Q is false positive. Here we can see that there is a chance of false positive too. So, this is a main challenge in bloom filter design and there are methods to reduce false positive. Also, we

have queried element Z in filter but Z is surely not present that means it's a true negative. From, here we can see that in case of bloom filter there is a probability of false positive but surely there is no false negative.

If the Bloom filter states a record is missing, we know it's missing and don't need to visit the disk. We're not sure if the record is present if the Bloom filter indicates it is. To find out, we'll have to query the database. When a filter reports something as present when it isn't, this is known as a false positive. False positives should be kept to a minimum in order for the program to function properly. Bloom filter improves performance when the majority of the searched records are missing from the database.

2.2.1 Parameters of the Bloom filter

Filter size **a**, number of hash functions **c**, and number of elements to be inserted **b** are the three main parameters. Use greater **a** and **c** numbers with a tradeoff for fewer false positives. The higher the **a**, more memory is consumed. The higher the **c**, the more computing is required. Because a higher **c** sets more bits, there is a c_{opt} number beyond which false positives start to rise. False positives increase as more items are added beyond the design limit. We should linearly scale m as n increases to ensure a stable false positive probability. In fact, the bit-to-element ratio is constrained by the following formula: **a / b ⩾ 1 / ln(2).**

2.2.1.1 Probability of False positivity

Let **a** be the bit size array, **c** be the number of hash functions and **b** be the number of expected elements to be inserted in the filter, then the probability of false positive p can be calculated as:

$$P = (1- [1 – 1 / a] ^{c.b}) ^{c}$$

2.2.1.2 Size of Bit Array

If expected number of elements **b** is known and desired false positive probability is **P** then the size of bit array **a** can be calculated as:

$$a = -\frac{b.ln.P}{(ln2)^2}$$

2.2.1.3 Optimum number of hash functions

The number of hash functions **c** must be a positive integer. If **a** is size of bit array and **b** is number of elements to be inserted, then **c** can be calculated as:

$$c = \frac{a}{b}\ ln2$$

2.2.2 Hash Function to chose

Bloom filters should utilize an independent and evenly distributed hash function. They must be as quick as feasible. Murmur, the FNV series of hash functions, and Jenkins hashes are examples of fast, non-cryptographic hashes that are sufficiently independent. Bloom filters rely heavily on hash generation. Bloom filter becomes slower as the number of hash functions **c** increases. Despite the fact that non-cryptographic hash functions do not provide any guarantees, they do boost performance significantly. Cryptographic hash functions provide stability and assurance, but they are time-consuming to compute. One developer showed how replacing MD5 with MurmurHash brought performance improvements [6].

In our model, we have used Secured Hashing Algorithm (SHA) as our hashing function. In our model, we have used SHA -224 hashing with bloom filter to protect the provenance information in bloom filter as suggested by [2]. Experiment done by the author shows that SHA -224 produces best output and in quick execution time. As opposed to our model, [7] has used SHA – 1 as a hashing algorithm.

2.2.3 Bloom Filter Application

Bloom filter is widely used in the applications that we use in daily basis like Google search engine, Google Chrome, Medium (American online publishing platform), and many more web applications. Bloom filter allows for quick searches, privacy preservation, content synchronization, and duplicate detection in databasebased applications. In a streaming application processing 17 million events per day per partition, Medium utilizes the Bloom filter to deduplicate recommendations. At scale, the Bloom filter was utilized to deduplicate events. The filter required 108 GB divided into 1024 partitions. Reads were 20x quicker than writes, while writes were 3x faster. The filter is used by the Chrome browser to represent a list of dangerous URLs. When a user requests a URL, if the filter indicates that the URL is likely to be secure, the request is sent to a server for verification. Bloom filters could be used in a web application to keep track of users originating from a given city viewing a webpage. [1].

Applications of Bloom Filters in Network Security can be widely found. Bloom filter is utilized in network-related applications such as peer-to-peer networks, resource routing, packet routing, and measurements. Bloom filters are used by Content Delivery Networks (CDNs) to avoid caching files that are only seen once. Author has discussed about the application of bloom filter in wireless networks for authentication, privacy preserving, firewalling, node replication detection, misbehavior detection and many more [8]. In our model too, we have used bloom filter for provenance encoding, decoding, collection & verification where confidentiality, integrity and freshness is achieved.

2.3 AODV Protocol

Routing protocols play a critical role in ensuring that communication between source and destination nodes is uninterrupted and efficient. The choice of a decent routing protocol has a significant impact on a network's performance, service, and

dependability. Wireless sensor networks and ad hoc networks must use round-free protocols. WSN routing protocols are classified in a variety of ways. The routing protocol is a method for selecting an appropriate path for data to flow from source to destination. While selecting the route, which is dependent on the type of network, channel characteristics, and performance metrics, the procedure confronts numerous obstacles. The data collected by the sensor nodes in a wireless sensor network (WSN) is relayed to the base station, which connects the sensor network to other networks (such as the internet), where it is processed and appropriate action taken.

Single-hop communication is used in very small sensor networks when the base station and motes (sensor nodes) are so close that they can communicate directly with each other. Sensor nodes in multi-hop communication not only create and transport their content, but also act as a conduit for other sensor nodes to reach the base station. Routing is the process of finding an acceptable path from a source node to a destination node, and it is the network layer's major responsibility.

We have already discussed about the challenges of WSN in Section 2. Considering the fact about low energy consumption, low processing cost, low memory, we have decided to use AODV [9] as our routing protocol which is reactive protocol and it performs better in throughput and end to end delay.

The ad hoc on-demand distance vector (AODV) protocol is a request-response mechanism. AODV is designed for mobile networks with little infrastructure. For route formations among network nodes, it uses the on-demand routing mechanism. When a source node wants to guide data packets, a single path is established, and the pre-set route is maintained for as long as the source node requires. This is the reason why we call it On-Demand.

The AODV routing protocol directs packets between wireless ad-hoc network mobile nodes. AODV allows mobile nodes to send data packets to a required target

node via neighboring nodes that are unable to openly connect. Routing table material is swapped between neighbor nodes on a regular basis and is prepared for unexpected updates. AODV uses broadcasting to find the best route via flooding mechanism. Route Request (RREQ) and Route Reply (RREP) steps are used in AODV to discover router and these request are relayed over the network to find the best path.

3. LITERATURE REVIEW

A wireless sensor network emerged as result of continuous research of network technology. It's a new idea, a vision for the future in the sphere of communication. So, a wireless sensor network is a network made up of a few to many small nodes having sensors and communications capabilities for transmitting and receiving data. The data collected by the sensors is sent through a network from one node to the next until it reaches the base station. With the help of this technology, such systems will be able to generate massive amounts of data and findings that can be analyzed in real time. This could lead to a new era of real-time monitoring and control of processes, something that was previously impossible without the intervention of people and complex equipment in some circumstances. Another benefit of using wireless technology is the cost savings associated with cabling deployment in current systems, as well as the ability to perform measurements in inaccessible locations.

Researchers and scholars are continuously working on improvement of security issue of WSN. Some of the important research in upgrading the confidentiality, integrity of data produced at sensor nodes will be discussed in this work.

“A Lightweight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks” has been discussed by Salmin Sultana [7]. In this paper authors have proposed a lightweight secure scheme to transfer the data securely in WSN. In this method “In-packet bloom filter” is used for encoding data at each sensor nodes i.e. intermediates nodes. Authors have discusses about confidentiality, integrity, freshness as well as provides protection against packet drop, loss, and replay attacks in the paper. The strength of this method is lightweight, effective and scalable. But this paper lacks the detection of the multiple malicious sensor nodes.

“Cluster-Based Arithmetic Coding for Data Provenance Compression in Wireless Sensor Networks” has been proposed by Qinbao Xu [10]. Wireless Sensor network consists of a large number of nodes which are deployed using distinct topology. In this paper, they have proposed a layered clustering management method for wireless sensor networks. As the

multilayered clustering model is used, provenance can be encoded independently and final encoded provenance consists of sets of different clustered layers. When base station obtains the provenance it decodes data from the top most layer. When it finds the malicious data in any cluster it stops decoding. This method also uses the local probabilities to encode provenance which has a higher compression rate. This clustered model is best suited when there is provenance of large size. This mechanism helps to increase the efficiency of data obtained at base station.

"Secure Provenance in Wireless Sensor Networks- A Survey of Provenance Schemes" is the paper proposed by Khizar Hameed and team of University of Management and Technology, Sialkot, Pakistan [11]. This paper discusses about the variety of provenance schemes used in the field of WSN research. In this paper, they have performed a comparative analysis of various secure provenance schemes based on general and security aspects. From this paper also, we can conclude the use of provenance based Bloom Filter proposed by [11] betters and can be used in our research.

"A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation" has been proposed by Cong Pu & Sunho Lim. s [12]. They suggest SCAD, a light-weight countermeasure to a selective forwarding attack, in which a randomly selected single checkpoint node is placed to detect malicious node forwarding misbehavior. To rapidly recover unexpected packet losses owing to forwarding misbehavior or poor channel quality, the suggested countermeasure is combined with timeout and hop-by-hop retransmission techniques. They also ran comprehensive simulation studies to evaluate performance and compare it to existing CHEMAS and CAD systems. The results demonstrate an increase in efficiency, a decrease in energy consumption, a lower rate of false detection, and a higher rate of successful drops. The limitations of this paper is that no any discussion on, changing the route of data packet after the malicious node is detected, is done.

"Design of A Secure Scheme employing In-Packet Bloom Filter for Detecting Provenance Forgery and Packet Drop Attacks in WSN" proposed by Rohit D. Hedau [13]discusses the

use of in-packet bloom filter for provenance forgery detection. The provenance scheme has been able to detect packet drop attacks staged by malicious data forwarding nodes. And the proposed scheme scores over the existing schemes in terms of throughput, Energy consumption, the packet drop ratio and the packet delivery ratio.

Fan Ye, Haiyun Luo, Songwu Lu, Lixia Zhang in their work "Statistical En-route Filtering of Injected False Data in Sensor Networks" have presented a Statistical En-route Filtering (SEF) mechanism that can detect and drop such false reports [14]. Multiple keyed message authentication des (MACS), each created by a node that detects the same event, are required by SEF to confirm each sensing report. Each node along the way verifies the correctness of the MACS as the report is forwarded, and those with incorrect macs are dropped at the earliest points. The sink filters out any residual bogus reports that slipped through the en-route filtering. Through collective decision-making by many detecting nodes and collective false-report detection by numerous forwarding nodes, SEF uses the network scale to determine the truthfulness of each report. Their analysis and simulations show that, with an overhead of 14 bytes per report, SEF is able to dmp % 90% injected false reports by a compromised node within 10 forwarding hops, and reduce energy consumption by 50% or more in many cases.

"Forgery and Packet Drop Detection Using Bloom Filter Mechanism in Wireless Sensor Network" proposed by Shruthy H.N. discussed Forgery and Packet Drop Detection mechanism with multiple consecutive malicious nodes [15]. Not many of the researches have been done on multiple malicious nodes detection. In this paper too, the limitation is that accurate forgery detection in case of multiple consecutive malicious nodes has also not been done.

4. METHODOLOGY

4.1 System Model

The primary goal of this thesis is to detect malicious nodes in wireless sensor networks. Our system gives an overview of how detection of malicious node in network is done. The Figure 4 is a graphical view of our system architecture for Malicious Node Detection System.

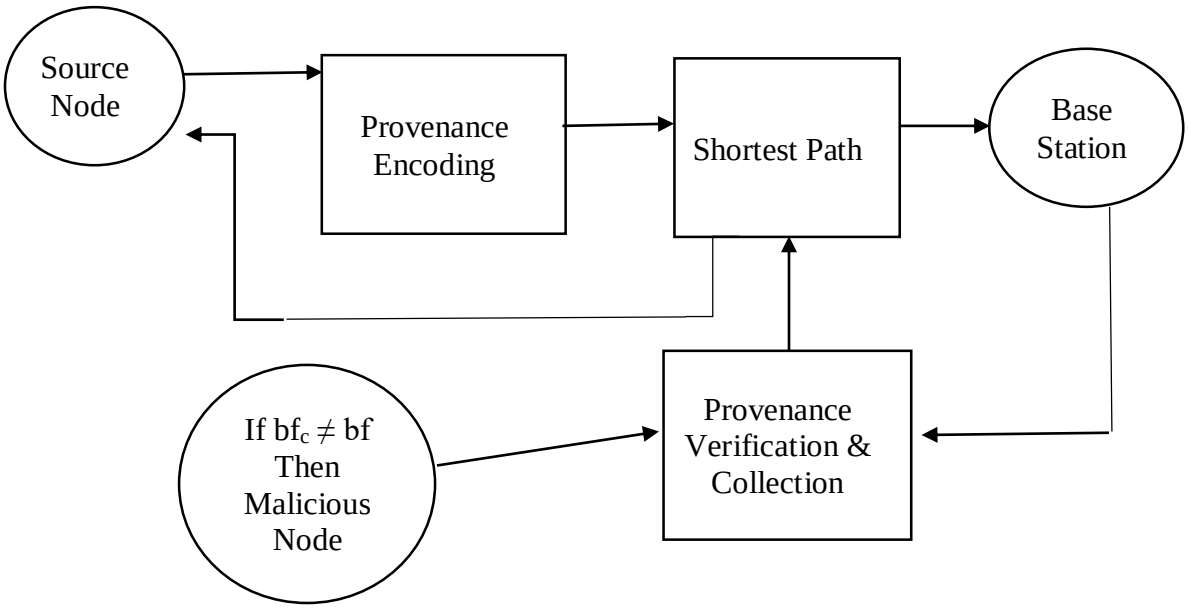

Figure 4: System Architecture

The provenance based scheme is shown in figure 4 to find the compromised node and to ensure the safety of the data packet in WSN. In our work, the provenance information is embedded into the bloom filter. The provenance information includes node name, port number, neighboring node and path cost. The information is provided to a data packet and all the nodes in WSN. The base station collects the data packet and regenerates the information and checks whether any of the nodes is compromised by the intruder or not.

Data provenance is an excellent way for assessing data trustworthiness because it summarizes the history of ownership and the actions performed on the data. Data

provenance allows the BS in a multi-hop sensor network to track the source and forwarding path of an individual data packet. The restricted storage, energy, and bandwidth limits of sensor nodes present significant hurdles in recording provenance for each packet. As a result, a lightweight provenance solution with little overhead is required. Also, security concerns including as confidentiality, integrity, and provenance freshness must be addressed. Our goal is to create a provenance encoding and decoding system that meets these security and performance requirements.

We describe a provenance encoding scheme in which each node along a data packet's trip securely embeds provenance information in a Bloom filter, which is then sent along with the data. After receiving the packet, the BS fetches and verifies the provenance information. We also created a provenance encoding scheme enhancement that allows the BS to determine if a packet drop attack was carried out by a hostile node.

4.1.1 Network Model

WSN consists of multiple nodes such as sensor nodes, intermediate nodes and base station. We consider a multi-hop sensor network with a number of sensor nodes and one base station in our model. Sensor nodes are stationary and do not move, however routing paths may alter over time as nodes fail or as attacks occur. Each node sends information to the base station about its neighbors. Each node is given a unique identification nodeID and a cryptographic key K_i by the base station. During provenance encoding, a set of hash functions is also broadcast to all nodes.

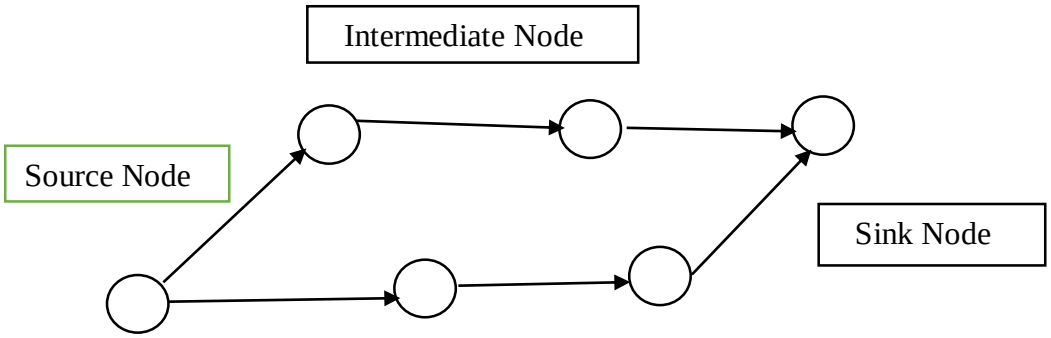

Figure 5: Network Model

Figure 5 shows the network model of our system. Here we have used mesh topology and all the nodes are connected to each other using some common links. We use AODV protocol to communicate between different nodes in a Network. Here, intermediate nodes can be legitimate node or can be a malicious packet dropping node.

4.1.2 Data Model

Sensor nodes of WSN is used to collect the data. Our scheme is to securely transmit this data to base station. Data generated by each sensor nodes is sent to base station using multi hop transmission. A data path of D hops is represented as < nl, n1, n2, ..., nD >, where nl is a leaf node representing the data source, and node ni is i hops away from nl. Each non leaf node aggregates its own data and provenance with received data and provenance.

Each data packet has its own packet sequence number, as well as data and provenance. The source node assigns a sequence number to each packet, and all nodes in a round use the same sequence number. Message Authentication Code is used to encode the sequence number (MAC).

4.1.3 Threat Model

We assume that base station and source node is trusted and all the intermediate nodes may be malicious. Malicious nodes may perform numerous attacks like passive attack and active attack. They can also make backdoor and perform traffic analysis.

Malicious nodes can deploy a few more other malicious nodes in the network or may compromise legitimate nodes. If a node is compromised then it can extract all necessary data & information, can also drop the packet or alter the packets. In our model, we consider packet drop attack in our network. Packet drop occurs when a compromised node or malicious node does not forward the packet but rather drops the packet during multi hop transmission. Our objective is to achieve confidentiality, integrity and freshness.

Different components of block diagram as discussed below:

4.2 Source Node

Sensor nodes are deployed at each stations in Wireless Sensor Networks (WSN) called as sensor nodes. These are actually hardware which sense environmental conditions like pressure, temperature, etc. These devices have relatively low battery life, low processing capability and low memory. Source nodes sense the data and send the data to the central station following numerous intermediate sensor nodes.

4.3 Provenance Encoding

In our method, the provenance is encoded to all the nodes and packet data. The provenance information consists of node name, port number, path cost, and the next neighboring node is uploaded to all the nodes. Then the base station receives provenance technique keeps the integrity of the data, which provides security and helps in detecting if any malicious data, is introduced by an intruder. Encoding of the provenance refers to providing each node with node name, port number, neighboring node, and the cost path. The provenance information is provided to all the nodes and each packet of data to be transferred.

The vertices in the provenance graph are created and inserted into the iBF for a data packet during provenance encoding. Each vertex is derived from a data path node and reflects the provenance record of the host node. A vertex's ID is a unique identifier (VID). The VID is generated for each packet using the packet sequence number (seq) and the secret key Ki of the host node. We use a block cipher algorithm to produce this VID in a secure manner. As

a result, the VID of a vertex representing the node ni for a particular data packet is computed as

$$VID_i = \text{generate VID } (n_i, seq) = EK_i (seq)$$

Where "E" is a secure block cipher.

When a source node creates a packet, it also creates a BF (known as ibf_0), which is initialized to 0. After that, the source constructs a vertex, inserts the VID into ibf_0, and sends the BF as part of the packet. Each intermediate node n_i performs data and provenance aggregation after receiving the packet. If n_j gets data from a single child, n_{j-1}, it combines the packet's partial provenance with its own provenance record. At last, when it reaches BS then total provenance if "ibf".

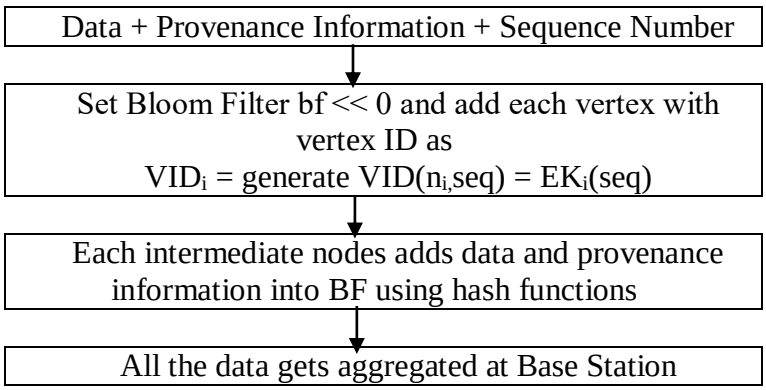

Figure 6: Provenance Encoding Algorithm

Provenance encoding is elaborated more in following diagram

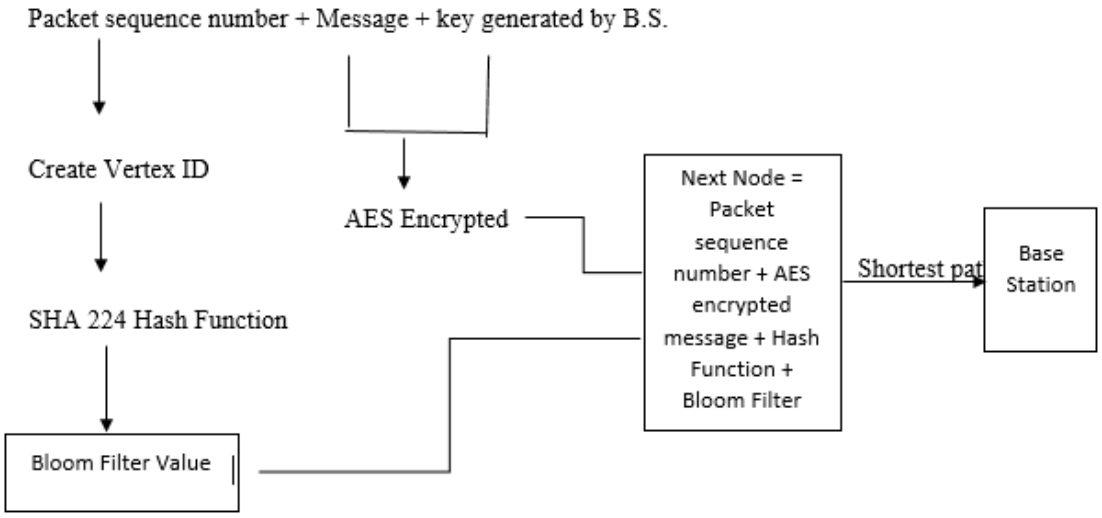

Figure 7: Provenance encoding in detail

Data is generated at source node. These data packet along with packet sequence number and provenance information should be transferred to the intermediate node. The data should be converted into cipher text using AES encryption. Key is sent to the node by base station as base station sends key to every node in the path of data transmission. Key along with packet sequence number generates vertex ID which is used to encrypt message using message authentication code. The encrypted value is passed to the SHA-224 hash function. SHA-224 hash function is suggested by Henry Nunoo [2] as it has faster execution time and consumes less energy which is mandatory in wireless sensor networks. Then, membership query of generated hash code is hashed into the elements of bloom filter. Initially, bloom filter is initialized to 0 in its space. If query matched into the bloom filter space then it is set to 1 otherwise 0. There may be possibility of false positive in bloom filter which we shall discuss through graph.

Now the encrypted message, packet sequence number and provenance information is sent to the next node using AODV protocol [16]. After receiving the information from previous node, it does data and provenance aggregation with the help of vertex ID generated in its

node. Again the data is aggregated in each intermediated node in routing path along with packet sequence number and provenance information.

When all the collected data and provenance information reaches destination node then base station records all the information and to ensure the integrity, confidentiality and freshness of data, it does provenance verification and collection on the traversed path. If received provenance information matches with generated information then provenance is verified otherwise an attack has occurred.

4.4 Shortest Path

When data packet encoded with provenance information is sent from source node all the way to base station then it has to pass to through several intermediate nodes. We will use AODV routing algorithm to transmit the message using shortest path. The weight cost of neighboring nodes of each intermediate node is calculated as the nodes are represented in the form of acyclic graph. Each node exchange its distance to its neighboring nodes using RREQ & RREP. We have already discussed about AODV Protocol in section background theories. This protocol works in finding the route from one node to another during data packet transmission.

4.5 Provenance Decoding

Provenance decoding starts when the base station collects the data packet. Base station conducts the provenance verification. The base station is aware of what path the message should follow. So it checks whether the correct path has been travelled or not. The provenance collection is necessary because the base station needs to recover the provenance information from the transmitted data packet. Base station finds out if any change in data packet has occurred by any malicious node or not. Decoded provenance is used to verify if the node is malicious or not.

4.5.1 Provenance Verification

We assume that the BS has P' knowledge about the packet's path. The Bloom filter BF_c is initially initialized with all 0s by the BS. After that, the BF is updated by producing the VID for each node in the path P' and inserting it into the BF. BS's assessment of the encoded provenance is now reflected in BF_c. The BS then compares BF_c to the received ibf to validate its impression. Only if BF_c equals ibf does the provenance verification succeed. If BFc differs from the received iBF, it means the data flow path has changed or a BF modification attack has occurred. To check whether the received path is modified or attacked then provenance collection is also done. The base station helps create the vertex for each node ni. The BS then uses ibf to run the vid_i membership query. The vertex is most likely in the provenance if the method returns true, indicating that the host node n_i is in the data path.

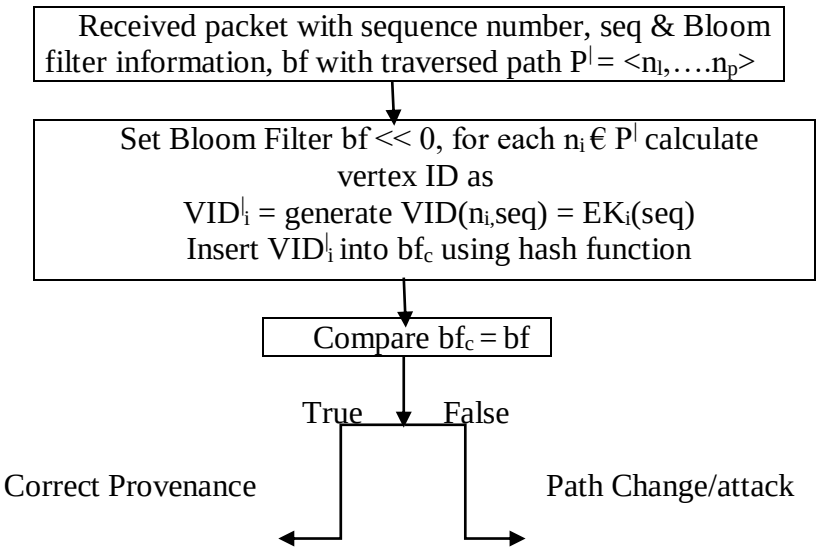

Figure 8: Provenance Verification Algorithm

4.5.2 Provenance Collection

Provenance Collection algorithm is shown in figure 9. Through ibf membership checking across all nodes, the provenance collecting scheme generates a list of probable vertices in

the provenance graph. For each node ni in the network, the BS creates the associated vertex. The BS then runs the vidi membership query with ibf. If the method returns true, the vertex is most likely in the provenance, suggesting that the host node ni is in the data path. There is a possibility of a false positive in bloom filter.

The provenance verification procedure is applied to the set of prospective candidate nodes once the BS has finalized it. This step is done to check between legitimate route modifications and malicious activity. If the verification is successful, we can conclude that the data flow changed naturally and that the path was correctly selected. Aside from that, there has been an attack.

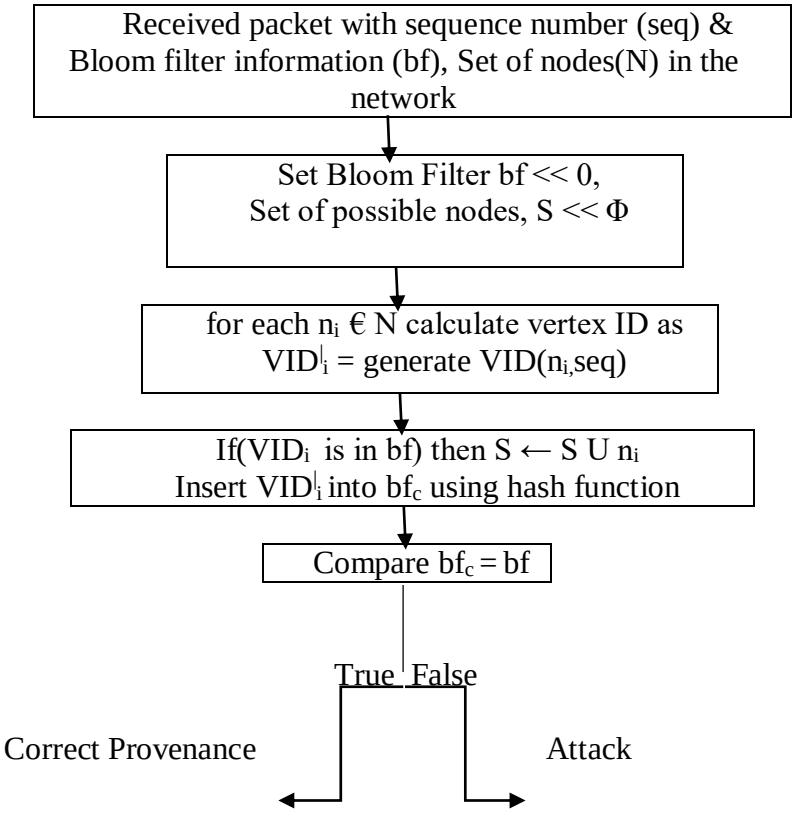

Figure 9: Provenance Collection Algorithm

4.6 Base Station

Base station is the destination node that collects all the sensed data from sensor nodes, which are deployed using wireless sensor network. Base station holds the data so that information can be extracted from data and correct decision making process can be executed. Base station after receiving packet with provenance encoding, decodes the packet in transverse path to determine any malicious node present using bloom filter based provenance decoding. Provenance encoding, provenance verification, provenance collection are shown in figure 6, 8 & 9 respectively. In all cases, base station is involved. Collected provenance in BF and data is aggregated in base station and in the process of provenance collection and verification, algorithm shown in figure 9 & 8 is executed.

4.7 Working of Bloom filter

Bloom filter is a data structure designed to confirm either the component is the part of the set or not. Burton Havard Bloom imagined the concept of the bloom filter in 1970. Bloom filter has been used and gradually modified time after time for the various network problems. Bloom filter is designed to provide with a unified and practical framework. Bloom filters are used as the large core memory for the applications which required the very large size of memory. Hence the use of bloom filter helps to reduce the unwanted use of memory which cost a very high price.

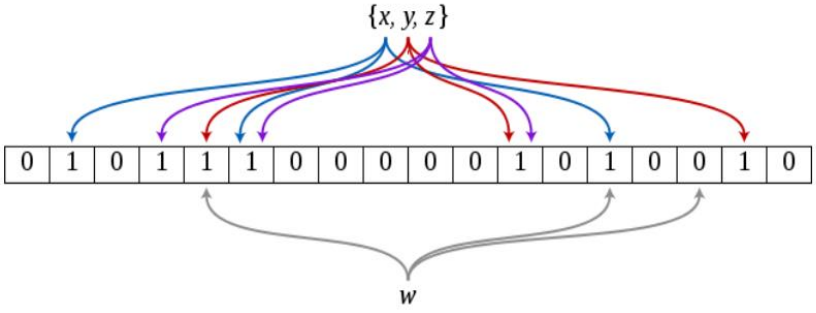

Figure 10: Bloom Filter Array

In the figure 10, 18-bit array is shown, where the set of x, y, z hashes are mapped into the elements of bit arrays. It can be seen that the hash function ‘w' is not in the set because it has been hashed to a one-bit array containing 0, which is not the part of the either of the precious sets x, y, and z. Initially, the empty bloom filter is set all to zero. ‘C' different hash functions are defined, and each of which hashes to the elements of array generating the uniform random distribution. Here the hash function ‘C' is constant which are much smaller in size than bit array. To check whether the elements are inset, elements are feed to **c** hash functions to get k array portion. If any of the element feed is 0 then it is not in the element or not in the set. In the above figure x. y, z, are hashed to 1 and w is hashed to 0 which is not the part of set x, y, z. Bloom filters have space advantage comparatively to the data structure to represent the sets. Bloom filter has a specific attribute, that false positive rate of the filter can be modified. The larger size of the bloom filter possesses less false positive comparative to a small one. We have discussed about bloom filter more in detail on section “Background Theories”.

Generalized step of our model

Step 1: Node initialize

Step 2: Topology Form

Step 3: Path selection

Step 4: Provenance encoding at each node

Step 5: Data transmission

Step 6: Data + Provenance collection at base station

Step 7: Performs provenance verification and collection to detect whether an attack has occurred or not

4.8 Sequence Diagram

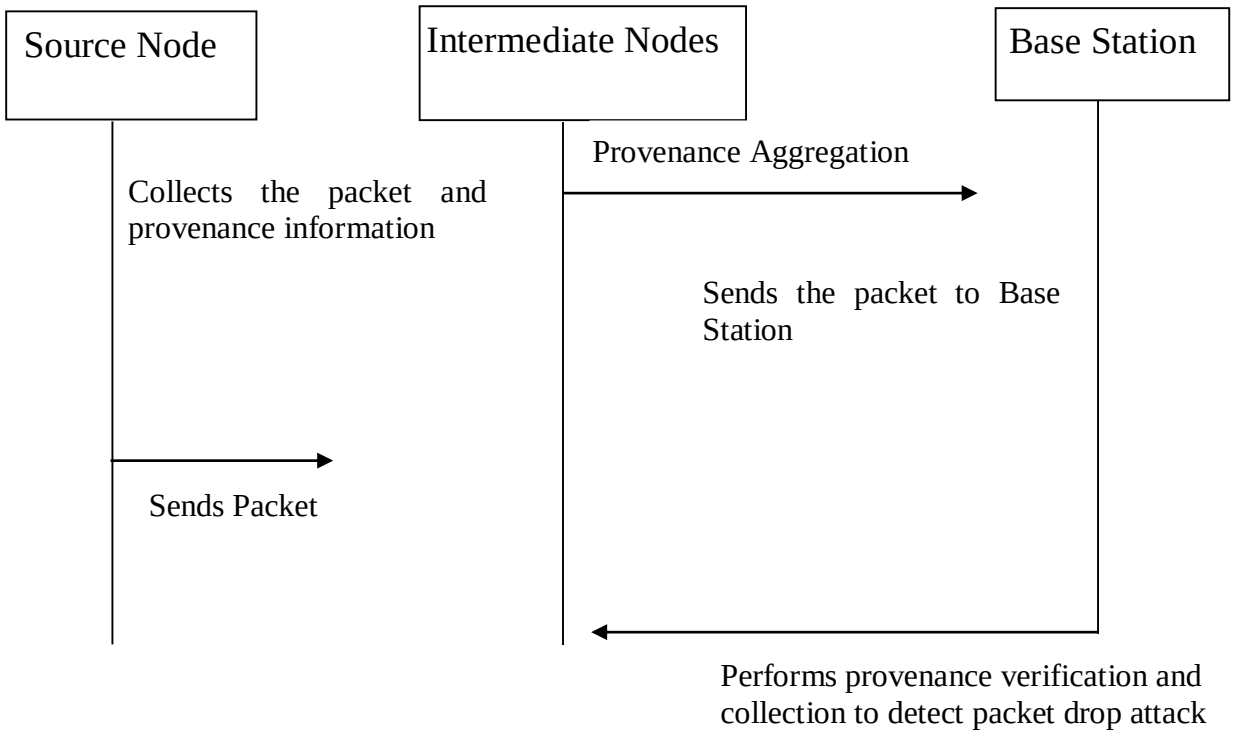

Figure 11: Sequence Diagram

The sequence diagram of our model is shown in figure 11. It consists of source node, intermediate nodes and destination node or base station. Packet is generated at source node. Thus generated packet will have to reach destination node through many intermediate nodes using provenance encoding mechanism. During process of transmission any intermediate node can be a malicious node which may alter the provenance information and also drop the packet so that correct message won't be sent to the base station which uses that data for decision making. So our model helps to detect provenance forgery and packet drop attacks in wireless sensor.

Base station will send each node a secret key which helps to calculate node ID for each node. This node ID is used to convert data to the cipher text and with the help of hash function, the generated code will be hashed into bloom filter which contains provenance information.

Initially this provenance is set to 0. At each node provenance is recorded using membership query of generated vertex ID with the provenance information using bloom filter and hash functions.

Each node in path of data transmission forwards the data to the nearest node which is in shortest path. When data reaches destination node, it collects the data and provenance record. Base station has the knowledge about traversed path of data transmission. Base station again calculates the provenance for each node in traversed path using equation of vertex ID. Sometimes the intermediate nodes may be shut down due to power issue or any path change of nodes. So to verify path change or attack, we perform provenance verification and collection algorithm. If there is no path change then base station knows about the attack in any of the intermediate node.

4.9 Detection of packet drop attack

Our system helps to securely transmit data from source node to destination node. We further used this scheme to detect packet drop attack in wireless sensor network and identify malicious node. In our work, each data packet contains acknowledgement of previously seen data packet which is useful in detecting packet loss attack. As the receiving node does not get previously sent data then provenance record generated will be different as provenance information now consists of data value, node ID, packet sequence number and sequence number of previously generated packet. This helps base station to localize malicious node and detect malicious node. We used multiple malicious node in our system to detect packet drop attack if any. Thus, in the extended scheme, any j^{th} data packet contains

(i) Unique packet sequence number (seq[j]),

(ii) Previous packet sequence number (pSeq),

(iii) A data value, and

(iv) Provenance.

The provenance record of a node includes (i) the node ID, and (ii) an acknowledgement of the lastly observed packet in the flow. The vertex ID, VID_i is generated as:

$$VID_i = \text{generate VID } (n_i, seq[j], pSeq_i) = EK_i\ (seq[j] \parallel pSeq_i)$$

Where, $pSeq_i$ is the n_i 's knowledge of the previous packet's sequence number in the flow. By putting VID_i inside the iBF, n_i updates the packet's provenance.

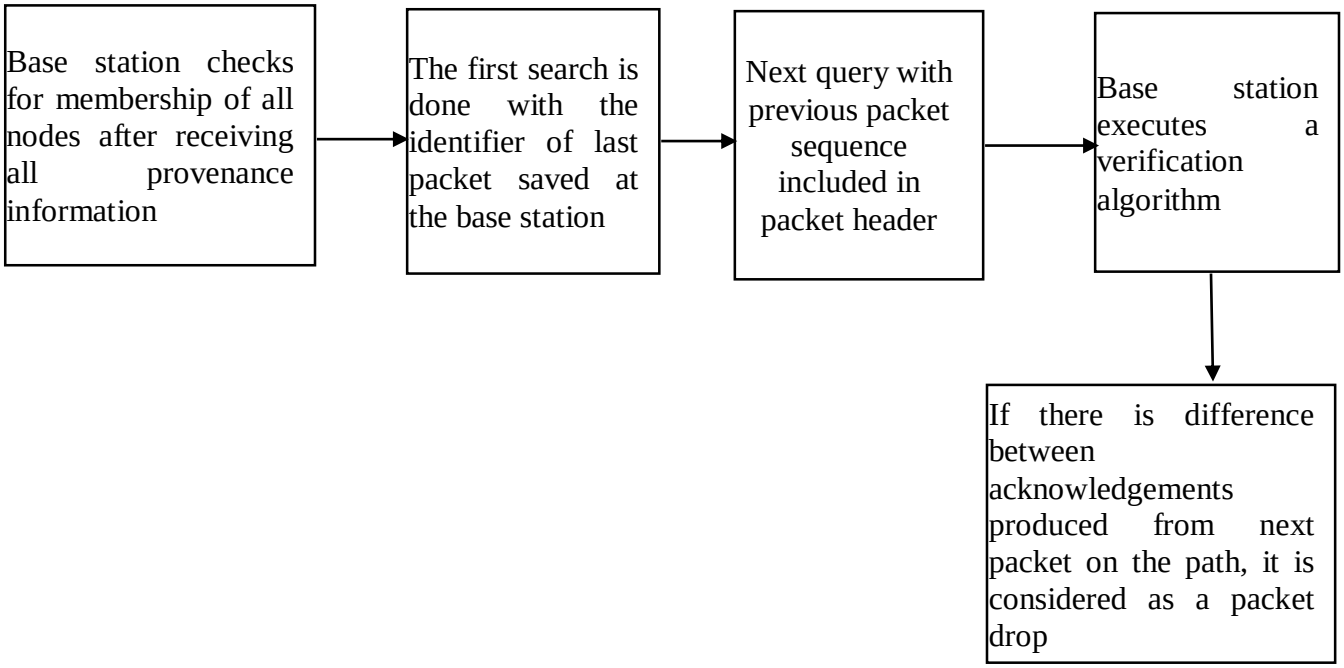

Figure 12: Packet drop process

Packet drop identification process is summarized in figure 12. Its detailed discussion is as follows.

It's worth noting that each data flow that passes through a node requires a perflow record to maintain track of the previous packet sequence. When a node ni processes/forwards the jth packet, it updates the pSeqi record of the accompanying data flow with the newly processed packet sequence, seq[j]. If a node gets a packet from a data flow for which it has no preceding packet information, it can use a special purpose identifier, such as 0, as the prior packet sequence pSeqi. When the routing path changes, this handles the circumstance where a new node in the path can use this specific identifier to encode provenance. In addition, if a node does not receive packets from a data flow for an extended period of time,

the flow's previous packet information may be removed to save space. As the node gets further packets from that flow, this flow-specific information can be updated and stored.

The BS performs provenance decoding after receiving the data packet, packet sequence number, and bloom filter. The BS, as well as the intermediate nodes, keeps and updates the most recent packet sequence number for each data flow. When the BS receives a packet, it extracts the preceding packet sequence (pSeq) from the packet header, downloads the last packet sequence for the flow (pSeqb) from its local storage, and uses these two sequences in the provenance verification and collection process.

Provenance Verification is identical to the fundamental architecture explained in Figure 6. When a packet is received, the BS first performs provenance verification. The base station knows i) Recent data flow path of the data packet ii) previous sequence number of packet in the path. The BS thinks that each node in the path saw and sent the same packet in the previous round, and that the packet's sequence number matches the one recorded at the BS. When pSeq and pSeqb do not match, the verification is bound to fail, indicating a possible packet loss and allowing the provenance gathering procedure to proceed without verification. The provenance verification is carried out in the same way as algorithm in figure 8, with the exception that the BS now uses following equation

$$VID_i = \text{generate VID}\ (n_i,\ seq[j],\ pSeq_i) = EK_i\ (seq[j]\ ||\ pSeq_i)$$

to create the VID for a node. The provenance collecting procedure is triggered when the verification process fails, indicating either a change in the data flow path or a packet drop attack. Provenance collection tries to recover nodes from encrypted provenance, confirm a packet loss, and track down the bad node that dropped the packet. It also differentiates between a packet drop attack and other potential attacks on the iBF. It's worth noting that, in the event of a path

modification, the new nodes can be quickly learned by performing an iteration of ibf membership testing across all nodes. During provenance encoding, each node in the path uses the same special purpose packet identifier as the preceding packet sequence and generates its VID as EKi (seq[j]||0). The BS's decoding scheme should execute ibf membership testing over all nodes to extract the new nodes in the path, with the VID for each node built using the previously supplied prior packet identification, as well as the nodeID and packet sequence number, seq[j].

Even in the case of packet loss, the provenance collecting algorithm can retrieve the nodes in a data channel. After getting the next packet (i.e. the $(j + 1)^{th}$ packet), the BS checks Even in the case of packet loss, the provenance collecting algorithm can retrieve the nodes in a data channel. The BS validates the membership of all nodes in the network within the iBF using a two-step method after receiving the next packet (i.e. the $(j + 1)^{th}$ packet). The first query uses the last packet sequence (pSeqb) recorded at the BS, and the second uses the preceding packet sequence (pSeq) included in the packet header. S1 and S2 refer to the sets of nodes discovered in the first and second steps, respectively. Let's call the BFs that were built with S1 and S2 BF1 and BF2, accordingly. The final Bloom filter, BFc, is a bitwise-OR of BF1 and BF2 that reflects the BS's perspective of the encoded provenance. The BS runs a verification procedure using the set of prospective candidate nodes S = S1 U S2 as input to check between the packet drop attack and any other iBF modification attacks. If BFc and the received iBF match, the verification is successful. In this case, we find that there has been a packet loss and that the path built on the set of nodes S is equivalent to path P. We were able to appropriately verify the origins as a result of this.

4.10 Equipment and Tools used

Table 1: Simulation Parameters

Network Simulator	NS 3.27
Operating System	Ubuntu 16.04
Network Diameter	500 x 500
MAC Protocol	IEEE 802.11
Routing Protocol	AODV
Traffic Type	UDP
Packet Size	512 Bytes
Node Mobility	Fixed
Number of nodes	2, 3, 4, ……., 500
Number of Malicious nodes	1, 2, 3, 4
Node Deployment	Random
Simulation Time	300 seconds
Initial Energy	1 kJ
WSN Topology	Mesh
Programming language	C ++ & Python

The summary of simulation parameters of our model is shown in Table 1. Here, we have used NS 3 as a network simulation being run on Ubuntu 16.04 operating system. Here, we have used network size of 500 x 500 m. IEEE 802.11 is used as a network interface to communicate with neighbor nodes. UDP packet of 512 bytes has been used to send data to the base station using AODV protocol. Researchers have suggested the use of UDP packet of 512 bytes in WSN [17]. Here, we have considered no network mobility that means we have used fixed network nodes. Network nodes are not mobile. Mesh topology has been used in our simulation environment. Mesh topology has already been discussed in background theory and network model. We have used different number of nodes in NS 3 i.e. varying nodes are used. Our main motive is to detect malicious nodes. Packet dropping nodes are

malicious nodes. Here, we have varied from 1 to 4 number of malicious node. Performance evaluation of different number of nodes on energy, throughput, and overhead are shown in result and discussion section. Simulation time used to execute the operation was approximately 300 seconds. Due to the use of encryption, hashing and provenance based encoding and decoding, simulation time was found to be larger. We have also calculated energy consumtion by each node in our model. Intitial energy of our sensor node was 1000 Joules (J). Effect of malicious nodes in energy was shown in figure 23. Script required to built and execute the model was written in C++ and python. Python was used for the user interface section. Gnuplot was used to show the simulation results. In our model, results were obtained by varying node density, number of malicious nodes, number of packets, etc. Detailed analysis of simulation results is shown in “Result and Discussion” section.

5. RESULT AND DISCUSSION

We have performed the secured transmission of data packet from source node to destination node using provenance based bloom filter. The network scenario was performed in Network Simulator 3 (NS-3). We used Ubuntu 16.04 as the operating system and also used NS3.27 as a network simulator. The programming language used are C ++ & python to run NS-3. The system was able to transmit the data packet from source node to base station using secured bloom filter mechanism where we have used AES-128 & SHA – 224 as an encryption algorithm. The obtained model was able to produce better performance based on Packet Delivery Ratio (PDR) & other factors.

We iteratively performed the simulation on NS-3 with varying number of nodes, packet size & Bloom filter size and hash functions. Our model performed well on Bloom filter of size 16 & 20 bytes. Bloom filter bit size was suggested by [18]. Author has compared bloom filter size with fpr and suggested the use of 16 and 20 byte filter. We used 4 hash functions in our model which we selected on iteratively running the simulation over 100 times.

On simulating the network in above scenario, following results were observed.

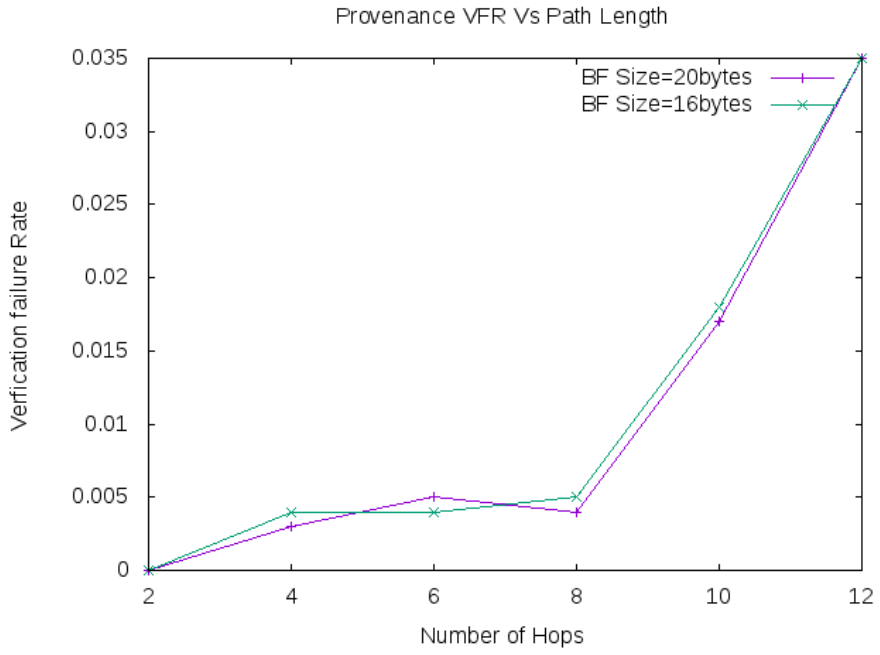

Figure 13: VFR vs Number of hops

We have seen from the figure 8 that verification at the base station fails if provenance information differ from the local information. This can happen due to Bloom filter Modification attack or data path change. Provenance verification failure rate (VFR) measures the ratio of packets for which verification fails. In figure 13, we can clearly see that VFR slightly increases as the number of hops or nodes increase in the network. The size of bloom filter varies. As the size of Bloom filter is 16 bytes and 20 bytes, the VFR is less. We have used SHA -224 hashing with bloom filter to protect the provenance information in bloom filter. The paper [7] used SHA -1 as hashing algorithm with bloom filter. We got the superior result in terms of VFR. The output of [7] is shown in Figure 14.

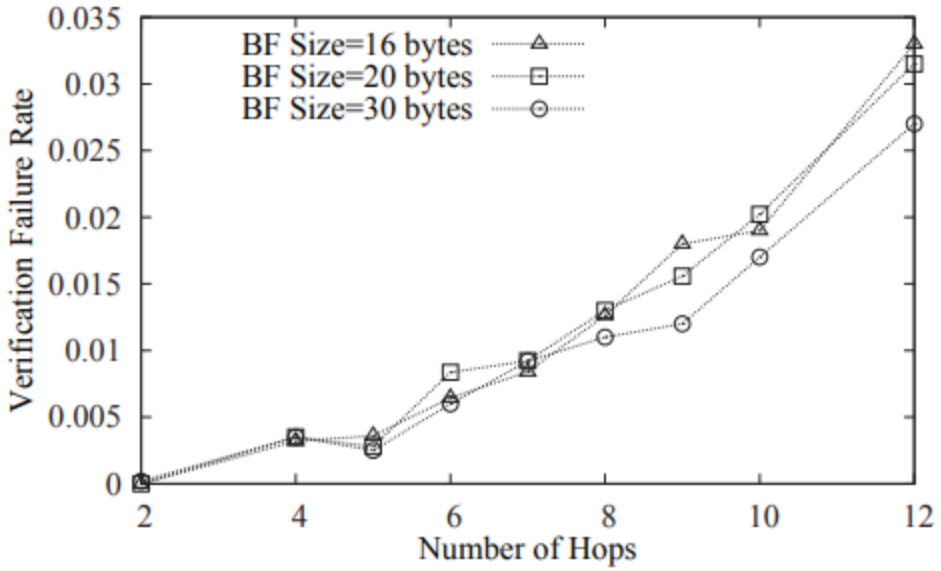

Figure 14: VFR vs Number of hops of reference paper

On comparing figure 13 & 14, we can conclude that VFR of our system performs better than VFR of [7]. On using bloom filter of size 16 & 20 gives the best VFR for securely transmitting data from source node to base station.

The main problem of bloom filter is false positive. As there is no false negative in bloom filter, we can have still chance of false positive. Our model helps in decreasing false positive rate as shown in figure 15.

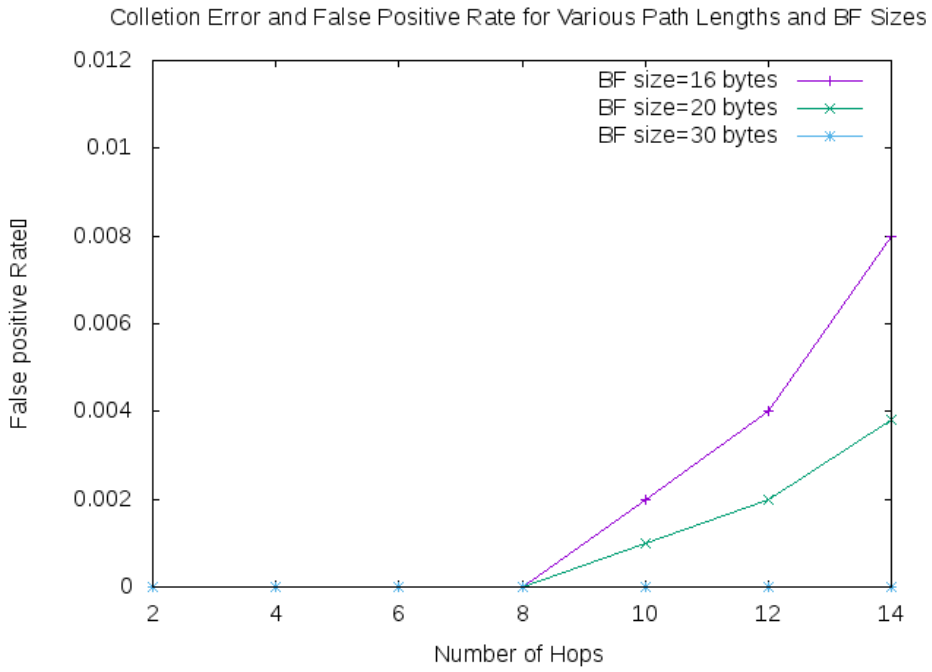

Figure 15: False Positive rate vs Number of hops

False positive rate is nearly zero till 8 number of nodes. If we go on increasing the number of nodes, there is slight increase in false positive rate as we can't assure 0 % of false positive when the number of nodes increase in WSN. The validation of our model is done with the reference paper of [7] shown in figure 16. Our model is superior than [7].

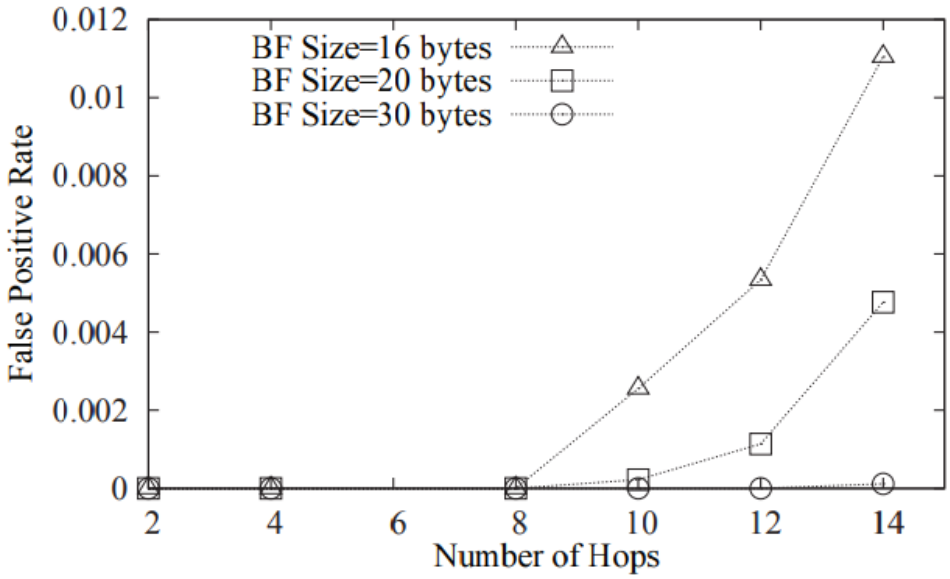

Figure 16: False Positive rate vs Number of hops of reference paper

Figure 16 shows the graph of False Positive rate vs Number of hops. We continuously varied the bloom filter size starting from 10 bytes to 40 bytes. Our model performed well when bloom filter is of size 16 & 20 bytes were used. We used the average round of 100 iterations and gained the result obtained in figure 15. On comparing our mode with reference model, we can find that our model is superior in terms of FPR when nodes increase after 10.

Verification failure rate (VFR) was compared with number of hops in figure 13. Here we compared VFR with number of packets. The result of simulation is shown in Figure 17.

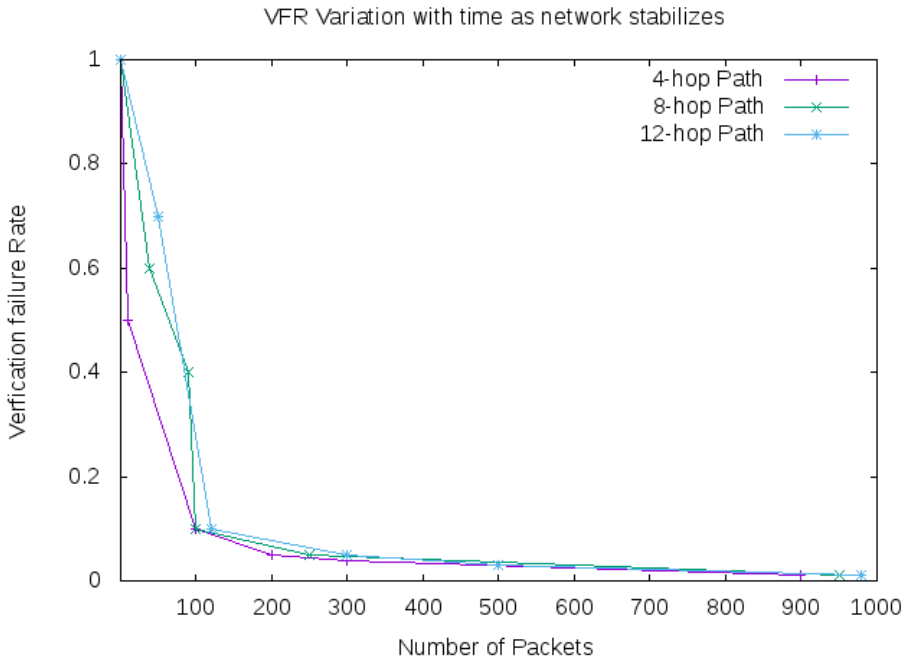

Figure 17: VFR vs Number of packets

Figure 17 shows when the amount of packet communications increases, VFR changes with time. As the network becomes more stable over time, the data pathways get less frequent, and the VFR approaches zero. As we can see from the figure 17, VFR is maximum with fewer packets, as the number of packets increases then VFR decreases exponentially. The route to the base station is determined using numerous nodes, so as the route is fixed then VFR will decrease. Our result is validated with the reference paper which has following output shown in figure 18. Figure 18 shows verification failure rate in terms of number of packets which shows that as the number of packets increase in the network then VFR gradually decreases and when number of packets increase beyond 100 – 200 then VFR is very small that it approaches zero (0).

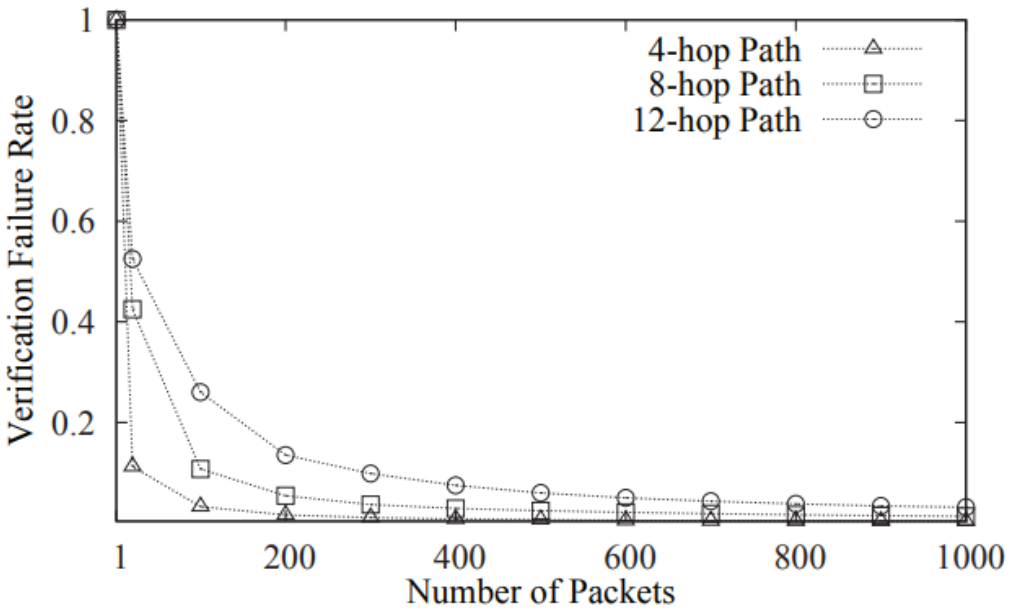

Figure 18: VFR vs Number of packets of reference paper

Our system was run on 4, 8 & 12 hop path, we got the result shown in figure 17. These hop paths were chosen iteratively, and the best path with minimum VFR is chosen. As we can see from figure 17 & 18, VFR vs number of packets is similar with both types. Here we, can conclude that VFR using SHA 224 & SHA 1 do not affect the VFR but previously we have discussed that there is slight improvement in VFR for number of hops in our model.

End to end delay is the time taken by the data packet to travel from source to destination. As the number of node increases in our network, end to end delay increases as well. Due to low routing overhead, end to end delay is not very large so the packet reaches to the destination in short interval of time. This is shown in figure 19.

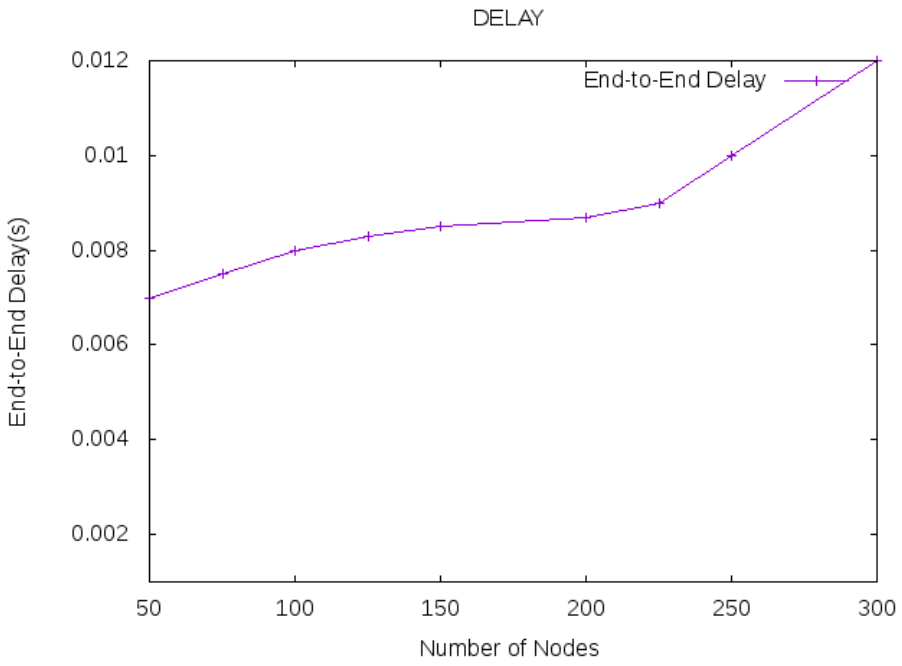

Figure 19: End to end delay vs Number of nodes

Figure 19 shows end to end delay vs number of nodes. Delay is measured in millisecond (ms). From figure 19, we can interpret that packet reaches from source to destination in less time. We have used AODV as a routing protocol. [9] suggests that end to end delay and routing overhead is outperformed by AODV protocol.

The problem with wireless sensor lies on throughput when data packets are sent from node to another node. The throughput of WSN is less compared to other wireless networks like Ad-hoc networks. Throughput is the maximum amount of data that can be sent from source to destination. Throughput plays a vital role in communication. Higher is the throughput, faster is the communication and vice versa. Now, throughput of our model is shown in figure 20 when there is no malicious node.

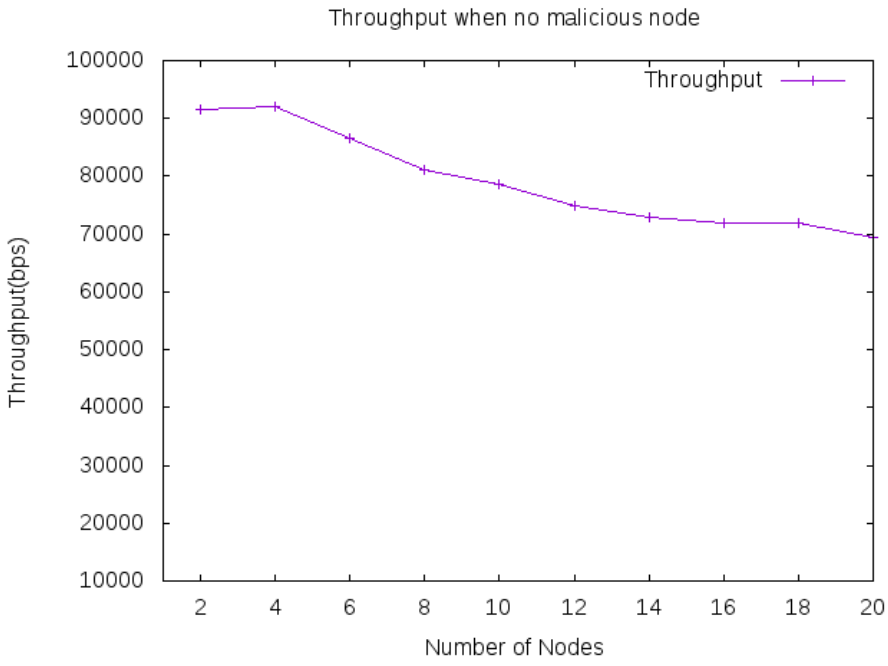

Figure 20: Throughput vs Nodes (When no malicious node added)

Figure 20 shows the throughput of WSN deploying number of nodes. Maximum throughput obtained with fewer number of nodes was 93.4 kbps. On increasing the number of nodes, there is slight variation in throughput due to end to end delay which is shown in figure 19. As we go on increasing number of nodes in WSN, we have seen that there is gradual decrement in throughput. Here, we did throughput analysis of our model in case of 16 & 20 byte filter. 16 byte filter gave the better result in throughput.

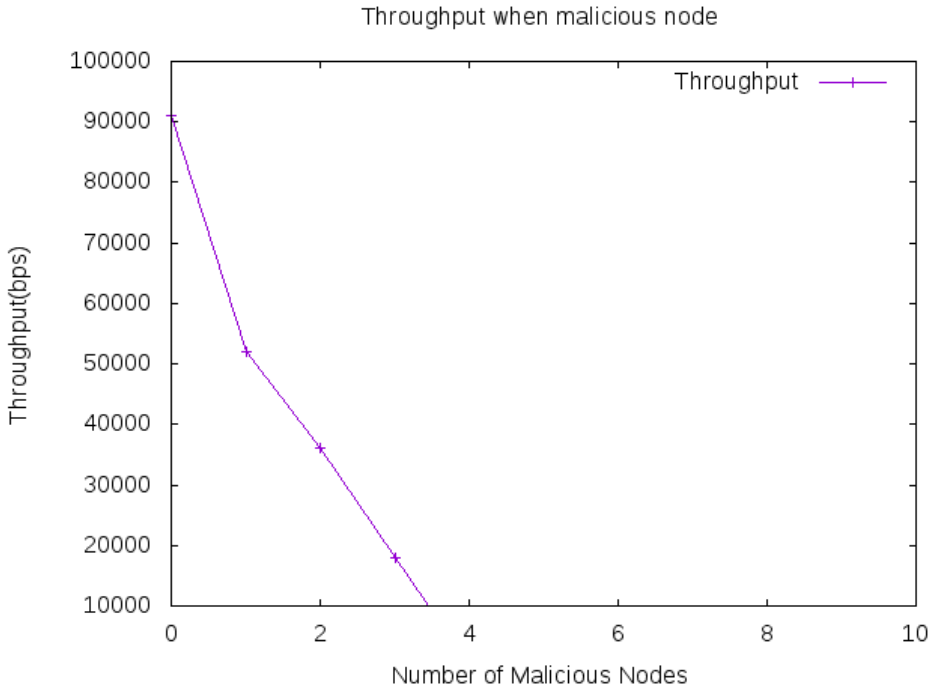

Figure 21: Throughput vs Nodes (When malicious node added)

Figure 21 shows the throughput of WSN deploying numerous malicious nodes. Maximum throughput obtained with no malicious node was 91 kbps. As, we deployed malicious node in our network, the throughput of our model decreased drastically. This shows the effect of malicious nodes in WSN. These malicious nodes deployed drops the packet and hence reduces the throughput of our network. Energy is also lost when dropping the packet. We will discuss about energy in latter part of our model. After deploying more than 3 malicious nodes in our network, throughput reduces to zero.

By using our model, we have successively detected malicious nodes in wireless sensor network using provenance based information in bloom filter.

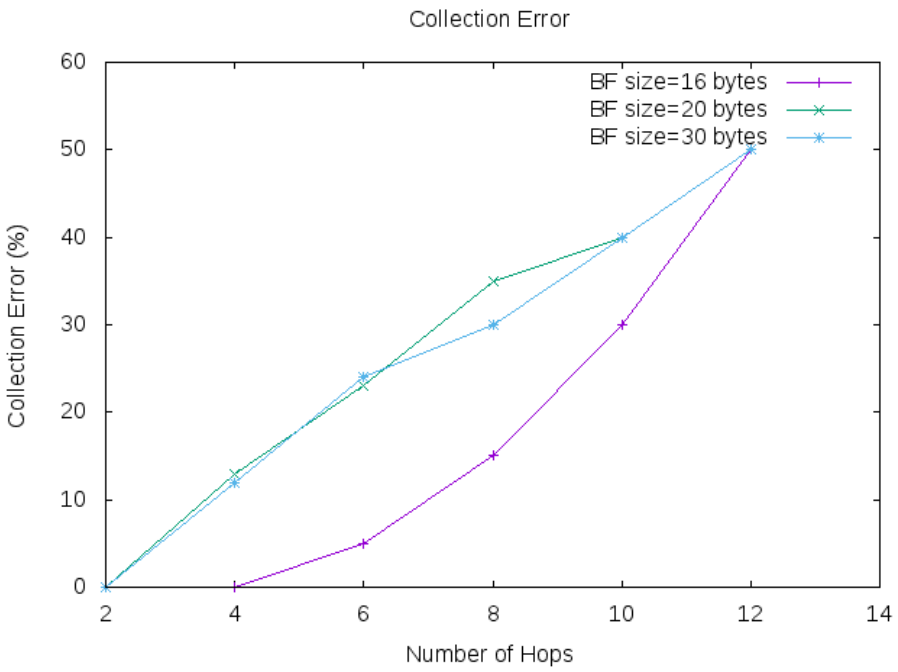

Figure 22: Collection Error vs Number of hops

The provenance decoding receives the provenance information from bloom filter. Provenance decoding process consists of provenance verification and collection phases. To find out the accuracy of our model, we measured provenance decoding error i.e. verification and collection error. We have already calculated verification failure rate (VFR) in figure 13 & 17. Note that, collection phase is executed when provenance verification fails.

Figure 22 shows collection error vs number of hops. Collection error occurs due to false positive rate. As false positive rate increases than collection error also increases. We can see from figure that collection error arises after 4 node topology but it increases more after 8 node topology. This is because of FPR shown in figure 15. We used bloom filter size of 16, 20 & 30 bytes. BF of size 16 outperforms 20 & 30. This shows that our model is light weight.

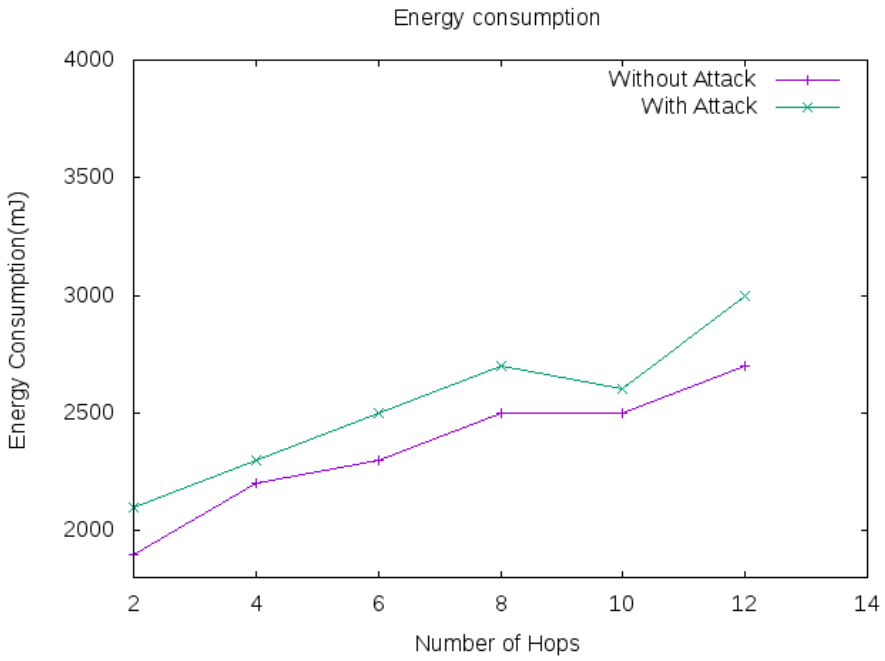

Figure 23: Energy Consumption vs Number of hops

Energy is a major constraint in WSN. We have already discussed about energy factor in background study part. Lowering the consumption of energy of each node is obtained using our model. We have used ns3 energy model to calculate energy of each sensor node. Figure 23 shows energy consumption of sensor node in case of normal operation and in case of attack. Energy is measured in milijoule (mJ). As the number of sensor nodes increase, energy of each node also increases. This is due to the fact that increase in sensor nodes results in increase in routing packet, data packet and provenance information as well.

In Figure 23, we have calculated the energy consumption in case of attack and no attack. As a result of packet drop attack, energy consumption increases in sensor node which results in delay of data packet as well. Average energy consumption of each node is shown in figure 23. Energy consumption for 2 node topology is 1920 mJ in normal mode and 2090 mJ in attack mode. As a result of increase in number of nodes, energy consumption also increases gradually.

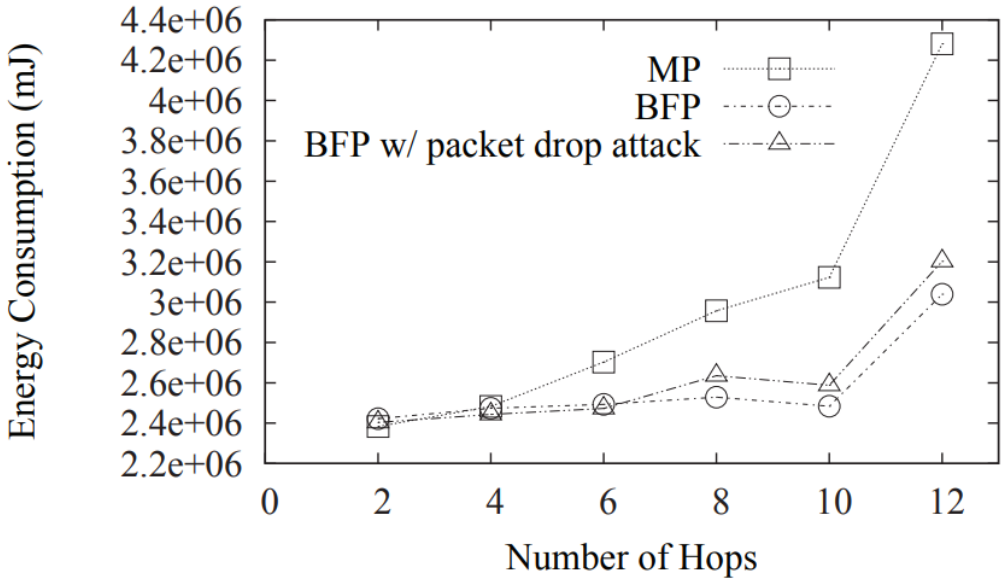

Figure 24: Energy consumption of reference paper

Figure 24 shows the energy consumption by nodes of wireless sensor networks discussed by [7]. Here, the author has compared the energy of model in case of normal mode and malicious mode. Energy consumption of 100 nodes are shown by author. Energy consumption gradually increases up to 10 node topology and when nodes increase beyond 10 then energy increases exponentially. On comparing Figure 23 & 24, we found that energy consumption by the nodes of our model is less compared to the reference paper. We got better results due to the use of energy efficient hashing algorithm [2] and energy efficient routing protocol [9].

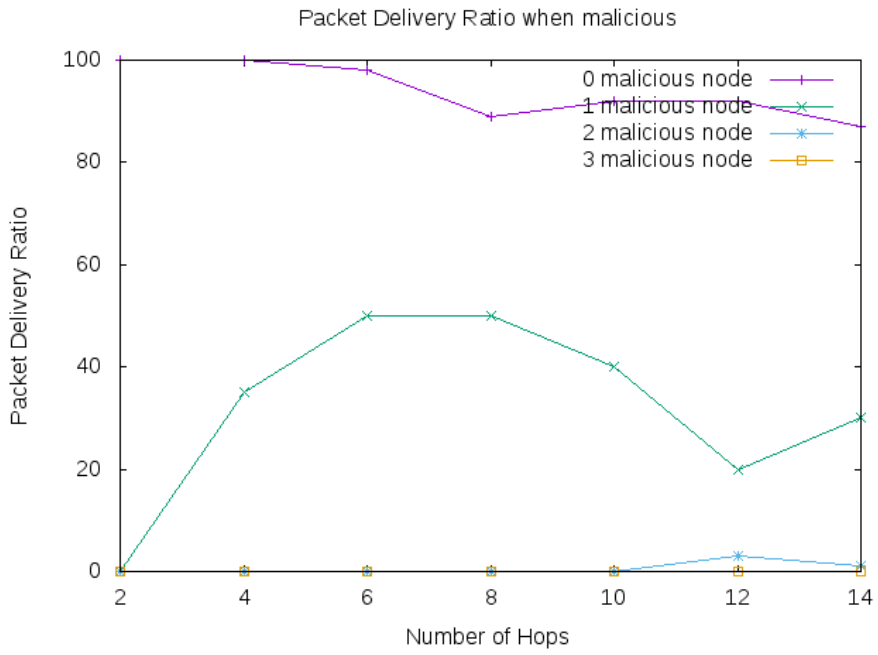

Figure 25: Packet Delivery Ratio vs Number of Hops

Packet delivery ratio (PDR) is the ratio between numbers of packet received to the number of packet sent by the sensor node. Ideally, PDR should be 100 % but due to different factors 100 % delivery ratio cannot be obtained. Figure 25 shows PDR for 0,1,2,3 malicious nodes. We can clearly see from figure that PDR is maximum in case of no malicious node. And as the number of malicious nodes increase, PDR also decreases and for more than 1 malicious node in topology, PDR approaches 0 i.e. whole packet is lost in between due to packet dropping nodes.

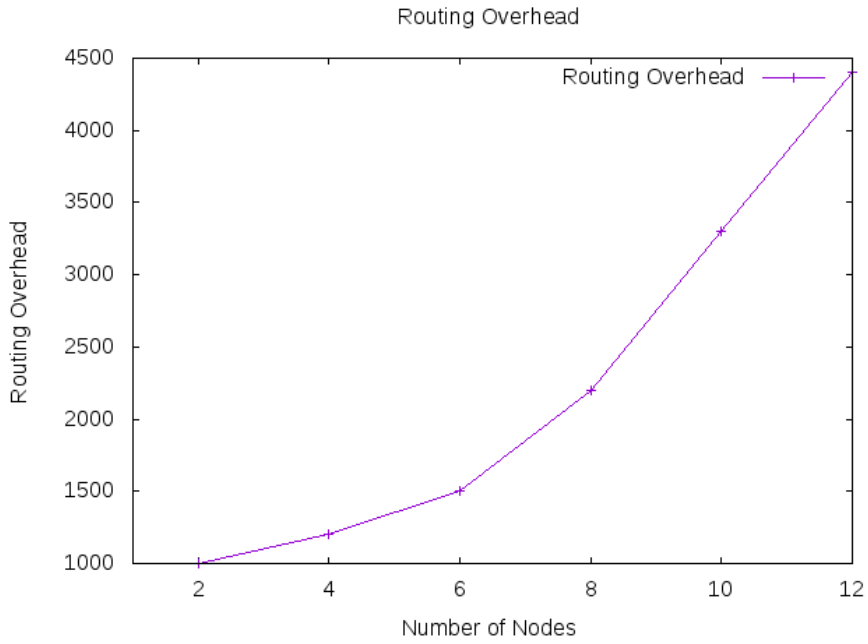

Figure 26: Routing Overhead in normal model

Routing overhead occurs as a result of the transmission of additional control packets required for proper data packet delivery. Routing and data packets should most of the time share the same network bandwidth, hence routing packets are regarded an overhead. This is referred to as routing overhead. The overhead of a good routing protocol should be minimal. Recent study [19] [20] [21] have shown that use of AODV is essential for reducing the overhead of the network. Figure 26 and 27 shows the performance of routing overhead in case of normal mode and in malicious environment respectively. Figure 26 shows that the routing overhead of AODV protocol in case of varying number of normal nodes. Here, the number of nodes increases then the routing overhead also increases. This is due to the fact that as the node density increases then the new routes have to be discovered to send HELLO packets and other meta data. Routing overhead increases exponentially to search for the new route in case of increase in network size.

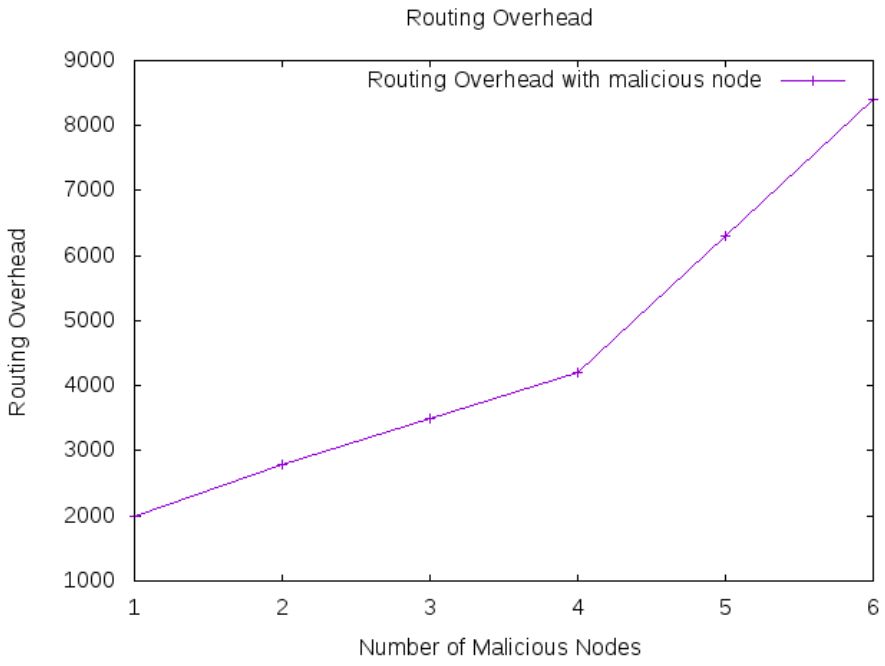

Figure 27: Routing overhead vs Number of malicious nodes

Figure 27 shows the performance of routing overhead in case of malicious nodes. Here we have varied the number of malicious nodes to identify routing overhead. In comparison to Figure 26, routing overhead of our model increases, this is because of the fact that new possible route has to be searched to send the data packet to the destination. On increasing the number of malicious nodes, routing overhead also increases to search for the possible new routes to the destination. AODV protocol performs better in terms of routing overhead in case of malicious nodes as well. As we go on increasing the packet dropping nodes in our topology, there is huge increment in routing overhead. This shows the effect of packet dropping nodes.

6. CONCLUSION AND RECOMMENDATION

6.1 Conclusion

A provenance based detecting of malicious packet dropping nodes in wireless sensor networks using bloom filter mechanism is presented in this work. We were able to securely transmit data packets from source node to destination using provenance based bloom filter. Here, we have used AES as encryption algorithm and SHA 224 as a hashing algorithm to hash the values into bloom filter. Using AES helps in maintain the integrity of the data packet and bloom filter helps in maintaining the freshness of the provenance information. We used our model to successfully detect packet dropping malicious nodes in wireless sensor network environment. Previous study have also shown the effectiveness of light weight bloom filter to detect malicious nodes. We have analysed our model against different parameters like Energy consumption, Packet delivery ratio, bandwidth, routing overhead, false positive ratio and end to end delay. Results shows that our model shows better performance in terms of different parameters. The use of bloom filter in network environment is increasing due to its low memory consumption especially in case of WSN. Hence we can conclude that our model was able to produce better results in detecting packet dropping nodes in wireless sensor networks using bloom filter mechanism with provenance information.

6.2 Limitation

There are some limitations of our work. The execution time of our model is high due to the computation in encryption and also due to the use of hashing algorithm. Our model has high routing overhead in case of increased network density. So, it is better to deploy this model in small sized network. This is due to the fact that AODV protocol performs better in small to medium sized topology.

6.3 Recommendation

Considering future work, we can enhance this model to detect other types of network attacks other than packet drop attacks. Bloom filter model can be used to detect attacks in IoT and Block chain models. Study shows that few works have been done with the use of bloom filter

to detect attacks in IoT, Software Defined Network (SDN) and other networking environment.

8. REFERENCES

[1] "Bloom Filter," Devopedia. 2020, [Online]. Available: https://devopedia.org/bloom-filter.

[2] "Comparative Analysis of Energy Usage of Hash Functions in Secured Wireless Sensor Networks," *International Journal of Computer Applications,* vol. 109, 2015.

[3] M. I. M.A. Matin, "Overview of Wireless Sensor Network," [Online]. Available: https://www.intechopen.com/books/wireless-sensor-networks-technology-and-protocols/overview-of-wireless-sensor-network.

[4] B. S. Rangstone Paul Kurbah, "Survey On Issues In Wireless Sensor Networks: Attacks and Countermeasures," *International Journal of Computer Science and Information Security (IJCSIS),* vol. 14, April 2016.

[5] S. N. a. S. K. B. Ripon Patgiri, "Hunting the Pertinency of Bloom Filter in Computer Networking and Beyond: A Survey," *Journal of Computer Networks and Communications,* vol. 2019, p. 10, 2019.

[6] M. Vicent, "Some perfomance tweaks," 2012. [Online]. Available: https://github.com/bitly/dablooms/pull/19.

[7] S. Sultana, "A Lightweight Secure Scheme for Detecting Provenance Forgery and Packet DropAttacks in Wireless Sensor Networks," *IEEE Transactions on Dependable and Secure Computing,* pp. 256-269, June 2015.

[8] M. A. Shahabeddin Geravand, "Bloom Filter Applications in Network Security: A State-of-the-Art Survey," *Computer Networks,* 2013.

[9] P. Singh, "Evaluation of Routing Protocols in MANETs with Varying Network," vol. 37, 2012.

[10] Q. Xu, "Cluster-Based Arithmetic Coding for Data Provenance Compression in Wireless Sensor Networks," *Wireless Communications and Mobile Computing,* 2018.

[11] K. Hameed, "Secure Provenance in Wireless Sensor Networks- A Survey of Provenance Schemes," *2017 International Conference on Communication, Computing and Digital Systems (C-CODE),* 2017.

[12] C. Pu, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal 12(1):834 - 842,* March 2016.

[13] R. D. Hedau, "Design of A Secure Scheme employing In-Packet Bloom Filter," *IJSRST,* 2018 .

[14] F. Ye, "Statistical en-route filtering of injected false data in sensor networks," *IEEE INFOCOM 2004,* 2004.

[15] N. D. Shruthy H.N, "Forgery and Packet Drop Detection Using BloomFilter Mechanism in Wireless Sensor Network," *International Journal of Engineering Trends and Technology (IJETT),* 2016.

[16] "Ad hoc On-Demand Distance Vector Routing," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Ad_hoc_On-Demand_Distance_Vector_Routing.

[17] S. K. A. K. Dwivedi, "Performance of Routing Protocols for Mobile Adhoc and Wireless Sensor Networks: A Comparative Study," *International Journal of Recent Trends in Engineering,* vol. 2, 2009.

[18] G. G. E. B. Salmin Sultana, "A Lightweight Secure Provenance Scheme for Wireless Sensor Networks," *18th International Conference on Parallel and Distributed Systems,* 2012.

[19] M. B. Luay Abdulwahid Shihab, "AODV routing protocol performance assessment for wireless sensor network scenarios," *International Journal Of Engineering And Computer Science,* vol. 10, no. 3, pp. 25292-25301, 2021.

[20] R. S. Anand Nayyar, "Simulation and Performance Comparison of Ant Colony Optimization (ACO) Routing Protocol with AODV, DSDV, DSR Routing Protocols of Wireless Sensor Networks using NS-2 Simulator," *American Journal of Intelligent Systems,* pp. 19-30, 2017.

[21] M. Y. E. R. F. S. Muh. Ahyar, "Comparison of Energy Efficiency and Routing Packet Overhead in Single and Multi Path Routing Protocols over S-MAC for Wireless Sensor Network," *UKSim-AMSS 6th European Modelling Symposium,* 2012.