A BLOCKCHAIN BASED FRAMEWORK FOR REPUTATION MANAGEMENT AND NODE MISBEHAVIOUR DETECTION IN WIRELESS SENSOR NETWORKS

by

Kartik Bhatia

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
May 2021

I dedicate this thesis to my parents for their inspiration and unconditional love and to my brother for his support and encouragement. Thanks for everything and for being part of my life. I love you.

Table of Contents

List of Tables

List of Figures

Abstract

With the growth of smart applications such as smart cities and smart farming, the importance of Wireless Sensor Networks (WSNs) is gradually being realized by many industrial enterprises. In particular, WSNs have shown enormous potential for being an interesting research area and in this decade, it is expected to grow manifold both in terms of applications as well as business revenues. WSNs consist of resource-constrained devices which are present in an open and unsecured environment, and this makes them vulnerable to both internal as well as external attacks. Internal attacks can affect the network's performance by increased energy consumption and introducing transmission delays. Consequently, this represents a critical security challenge for the deployment of WSNs.

Many researchers have proposed solutions based on trust management systems that proves to be an efficient way for detecting such attacks by enhancing trust relationships and data routing reliability. In this thesis, we extend the trust management system to include a distributed consensus mechanism based on blockchain which validates data packets originating from various source nodes. Additionally, a new algorithm is developed to estimate a node's reputation based on its historical energy consumption data. Reputation and trust are both crucial factors that characterize malicious behaviour in the network.

We have evaluated our proposed work with another existing trust model named Belief-based Trust Evaluation Mechanism (BTEM) and compared our results in terms of performance metrics after performing various simulation runs. The results show that there is a significant improvement in the detection rate and accuracy. Furthermore, we have shown that our framework fulfils important security requirements such as integrity, authenticity and confidentiality by analyzing it for various security attacks. Additionally, based on our analysis and findings, our framework can be used to detect any malicious activity in the routing process of a wireless sensor network.

List of Abbreviations Used

ADC	Analog to Digital Converter
AODV	Ad Hoc On-Demand Distance Vector
BS	Base Station
BSN	Body Sensor Network
CA	Certified Authority
CH	Cluster Head
CIA	Confidentiality Integrity Availability
DLT	Distributed Ledger Technology
DOS	Denial of Service
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
NS3	Network Simulator 3
PDOS	Path-based Denial of Service
PoA	Proof of Authority
PoW	Proof of Work
P2P	Peer to Peer
RPL	Routing Protocol for Low-power and Lossy Network
RTS	Request to Send
SNR	Signal to Noise Ratio
TXN	Transaction
UDP	User Datagram Protocol
USD	United States Dollar
UTXO	Unspent Transaction Output
WSN	Wireless Sensor Network

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Dr Srinivas Sampalli, for their guidance and support throughout my master's program. He always provided valuable inputs and encourages me to explore more in my research work. Thank you for guiding me and allowed me to indulge in research projects.

I would like to thank my committee members, Dr. Qiang Ye and Dr. Peter Bodorik for accepting to be part of the committee. I truly appreciate your time and suggestions to improve my thesis.

I want to thank my parents, Raj Kumar Bhatia and Isha Bhatia, and my brother, Sagar Bhatia, who has always been caring, loving and supporting me throughout my life. I am very grateful for giving me a golden opportunity to complete my further education. I would specially thank my friend Miheer Kulkarni for being supportive and helping me during the designing phase of my research. Finally, I want to thank God for giving me strength and all my family and friends for their love and support.

Chapter 1

Introduction

1.1 Introduction to Wireless Sensor Networks

With the emergence of wireless communications, Wireless Sensor Networks (WSNs) are becoming a growing field in computer science research. They are being used in several application areas, ranging from defence to commercial businesses. The WSN consists of numerous independent devices that allow us to collect data, monitor environmental parameters, and detect state changes. In the WSN, the two key components are the sensor nodes and the base station. The sensor nodes are interconnected using wireless communication techniques and consist of many components such as a transceiver, micro-controller and battery. The primary purpose of the sensor node is to sense environmental factors or physical properties, gather data and send it to the base station. Meanwhile, the base station, also called a sink node, receives data from sensor nodes and connects the network to the outside world [12].

1.1.1 Applications

Due to recent innovations in technology and networking, WSNs cover a broad range of industrial and environmental applications such as military, healthcare, agriculture monitoring, fire detection, bridge monitoring and environmental sensing [12]. Some of the major applications are discussed below:

Military applications These applications involve tasks such as area monitoring for security with wireless sensor networks. The main goal is to receive accurate information securely. The sensors are used to track and detect enemy movements which involve waking up the nodes from the sleep state and detecting intrusions from the enemy. Another usage of WSN by the military can be battlefield surveillance involving mobile sink nodes such as drones to collect data from the region of interest [25].

Health care applications The WSN is used in healthcare applications because they allow an efficient way to record patients data periodically [25]. The sensors are placed close to the patient's body for health monitoring and are known as Body Sensor Networks (BSNs). The BSN enables continuous monitoring of unattended patients thus, improving the efficiency of patient treatment. These sensors can be of different types such as wearable, implantable, and can be used to measure patients critical body parameters such as blood pressure, heart rate, etc. If these parameters show any abnormality, they can help in detecting early signs of diseases. The monitoring requires accurate, secure and robust data collection, including the privacy of data [25].

Industrial applications Wireless sensor networks have an important use in industrial applications. Some of the important functions handled by wireless sensor networks include machine health monitoring, machine fault detection, and much more. It allows industries to monitor machine performance periodically without investing in manual inspection, which helps companies save money and effort. Additionally, this also improves the production rate since faults are handled by sensor nodes which allow the company to respond immediately to any failure. Furthermore, WSNs can also be used to detect any water pipeline leakage caused due to cracks or blockages in the pipeline [25].

Environmental applications Due to rapid industrialisation, our environment is facing challenges caused by human activities such as the emission of greenhouse gases. WSNs can be used to take preventive measures in these cases, such as checking the quality of the air, monitoring the impact of forest fire and natural disasters [25]. Some applications need constant monitoring of the environment in agriculture, such as mushroom cultivation in an indoor area that requires controlled humidity to maintain the quality of produce. Some other applications like weather forecasting make use of environmental data.

1.1.2 Security Challenges

Many applications require reliable data delivery while preserving the security and privacy of the data. Due to sensors deployed in hostile environments and using wireless medium to communicate between them, these devices are susceptible to different attacks and can get easily compromised by adversaries. These adversaries can launch

insider attacks using the compromised nodes making security a challenging issue. Moreover, using the traditional approaches of encryption and decryption process is very resource-intensive. Thus, we cannot directly use these techniques because of resource constraints like limited energy and low computation capability [19]. To implement security mechanisms, several researchers have modified and developed various security techniques to ensure the WSN is secure against any malicious attack. However, many security challenges still need to be considered before proceeding ahead with developing various security models. Some of the key security requirements that are necessary for the wireless sensor network are data reliability, availability, integrity, and confidentiality.

1.2 Motivation and Objectives

WSNs have seen a growth in recent years due to the advancement of various sensor technologies and their integration into the Internet of things (IoT) applications. However, these networks are vulnerable to many security threats, such as internal and external attacks, because they operate in hostile environments. The adversaries can directly launch an attack at the network layer of the node. These attacks can prove to be detrimental to the performance of any application. Many critical applications like healthcare and defence are at high risk to these potential threats, as they can bring systems to a halt and cause significant economic losses as well.

Currently, the researchers are mainly concentrating on multi-path routing [7], secure data aggregation [34] [24], localisation [18] etc., while largely ignoring the security measures that need to be considered to prevent any malicious attacks during data transmission. During our literature review, we found that few papers [35] [19] have addressed the mitigation of routing attacks, while others [32] [14] have relied on traditional security mechanisms like key pair exchange and authentication. Literature survey has shown a lack of trust between the sensor nodes related to the exchange of information, which has led some researchers [26] [30] to propose trust and reputation-based security systems for detecting any anomalous behaviour of a node. Moreover, we also observe that these techniques do not validate the provenance of the data packets before predicting any maliciousness present in the network. Some solutions also propose to use blockchain at the cluster head levels with Proof of Work (PoW)

consensus that is infeasible for a WSN [35] [16].

Our main objective is to design a unique framework that detects malicious nodes present in the network by integrating decentralised blockchain technology and trust management systems to overcome the shortcomings of other existing work. Firstly, we consider two critical node parameters, namely, energy consumption and packet forwarding rate, to mitigate the lack of trust. In particular, we choose these parameters because both the energy consumption and packet forwarding rate provide useful insights into the behaviour of the node during the routing phase. These parameters are recorded periodically and utilised to compute the trustworthiness and reputation of each node. Secondly, the data packets are validated by a decentralised blockchain that contains records of identity, key and reputation maintained by several base stations in a distributed manner. Our work examines the potential for a combination of a trust model and the blockchain-based reputation that expects to make more accurate decisions about malicious nodes present in the communication network by working with the blockchain model to validate the data packets. Furthermore, through this work, we demonstrate how blockchain can be useful for low power and low computational capability systems such as wireless sensor networks. We also illustrate how our framework can detect various security attacks such as selective forwarding, packet modification, and Sybil attacks.

1.3 Contribution

A variety of solutions have been proposed to prevent and detect malicious attacks. Some of the proposed solutions rely on the trust model or utilise blockchain for storage and identity management. Some drawbacks of these solutions are that they do not compute a node's reputation based on historical data, and the majority of them evaluate malicious behaviour based on a single metric. Moreover, the existing solutions use a centralised mechanism to manage the trust information, resulting in a single point of failure. Most of these works fail to address security goals like data integrity, data availability, and data confidentiality. The major contribution of this work is that we propose a framework that integrates private ledger technology (blockchain) with the trust management models to produce a single security mechanism that provides traceability of the data packets in wireless sensor networks. This framework is used

to detect malicious nodes that use the historical trace of the message records for calculating the reputation before declaring a node as malicious. Some of the detailed contributions are mentioned as follows:

- We design a decentralised blockchain that allocates key pairs and enrollment certificates to sensor nodes.
- We validate sources and intermediate cluster head nodes before processing their data packet.
- We devise a non-linear function to aggregate historical confidence scores to compute a reputation score for each sensor.
- We also analyze how our framework achieved security goals like confidentiality, integrity and availability.
- We also achieve authenticity by registering the node identity in the blockchain before communication starts.
- The evaluation results reveal our framework has better performance than the existing solution BTEM.
- We apply Proof of Authority (PoA) as a consensus mechanism for block generation for low power sensor networks such as WSN.

1.4 Thesis Outline

The rest of the thesis is organised as follows: Chapter 2 provides the necessary background for understanding the thesis including the overview of cluster-based routing, security threats in WSN and blockchain. Chapter 3 reviews the literature survey on the recent work related to blockchain implementation and trust management models. Chapter 4 introduces the proposed reputation management framework using blockchain technology and also discusses its design, methodology and algorithms. Simulation results and discussion on its comparison with other model is given in Chapter 5. Chapter 6 present conclusions and future research directions.

Chapter 2

Background

2.1 Wireless Sensor Network(WSN)

The WSN is a distributed network of various tiny-sized and resource-constrained sensor nodes. These sensors have limited energy, low memory storage and smaller computation capability. The sensor consists of a sensing unit, power unit as an energy source, processing unit including analog to digital converter (ADC), memory, processor and communication unit containing transceiver [25].

Generally, the network consists of a set of cluster nodes and sink node which interacts with each other through a wireless medium typically via a single-hop or multi-hop process. The sensor's main task is to collect data and disseminate that information towards the sink node where the sink node receives all data from the sensor and provides this information to the end-user. These sensors are widely used in various applications, including tracking, measuring, and monitoring physical and environmental variables like temperature, pressure, etc.

One example of the application of WSN is smart farming where multiple sensor nodes are deployed all over the farm to gather data on factors such as humidity levels to monitor and detect changes in the state of the soil. This data is forwarded via the sensor nodes to the cluster head which further sends the data to the base station via multiple cluster heads. Once all the data is received at the base station, the collected information is analysed and an appropriate decision is made. Even though the network appears small, the WSN is still vulnerable to many security threats such as packet drop attack that can be detrimental to the overall network performance.

2.1.1 Sensor node architecture

The sensor nodes serve as the basic building blocks of WSN that perform sensing of any state changes. To perform these operations, the sensors consist of various units

like sensing, power, processing and communication unit that operate simultaneously to serve the purpose of parameter sensing in the nearby surroundings. In addition, a sensor node operates in different phases during a packet exchange like sleep, idle, receiving, transmitting etc. Figure 2.1 represents the architecture of a sensor node [31].

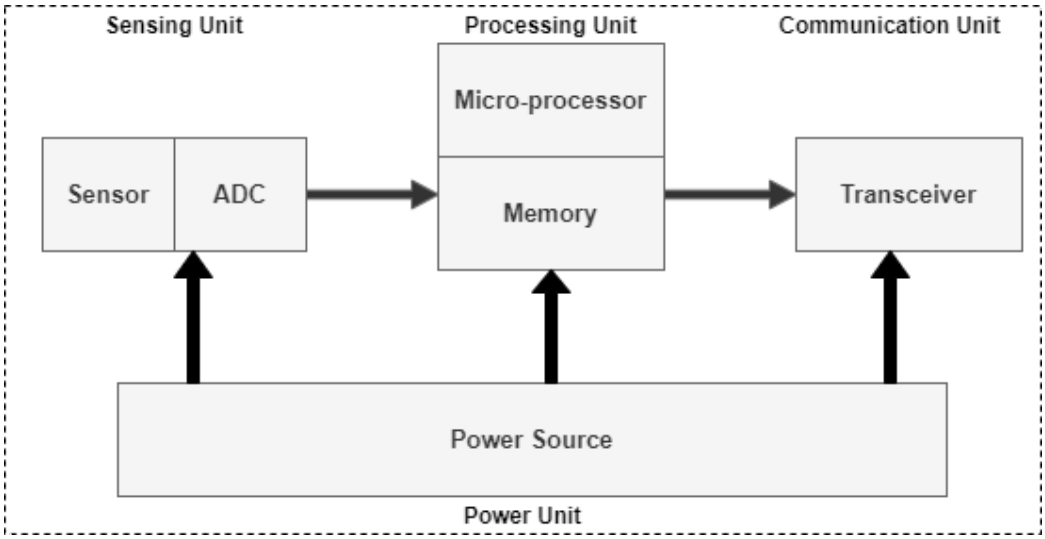


Figure 2.1: Sensor node architecture

The sensor node architecture consists of four major units:

Sensing unit One of the primary components of a sensor node is the sensor. This particular sensor is primarily used for detecting any physical changes observed in the surrounding environment and are equipped with the ability to sense parameters such as temperature, pressure, etc. It also consists of an Analog to Digital Converter (ADC) as data sensed is in an analog form that needs to be transformed into digital form for transmission over the wireless network.

Power Unit: Essentially, it is an energy source for the sensor node powered by a limited battery where each node's phase change is associated with a variable amount of power consumption.

Processing unit: It is the core module of the entire sensor node architecture which is responsible for collecting and storing the data in the appropriate format. It consists of memory which acts as a storage location for buffering data for a short period. Another component is the processor that analyses the data and performs necessary operations like packet forwarding through routing rules embedded in it.

Communication unit: Sensor nodes connect and communicate with each other through a transceiver on a wireless communication channel. Sensors mainly use radio frequency (wifi radio) to interact with other nodes.

2.2 Security Attacks in WSN

In recent years, computer network security has been facing severe challenges. A recent study estimates that 95 percent of security attacks are targeted against three industries, namely, defence, government services and small and medium-sized businesses. This has resulted in a loss of almost 3.9 billion USD per year across these sectors [21]. Furthermore, this has resulted in data compromisation and affected the users' privacy. The wireless sensor network has not escaped these attacks. The same study [21] suggests that approximately there is a 95 percent chance of security attacks in WSN. This can lead to a loss in data confidentiality and data integrity. These attacks can be classified into two types based on the degree of impact on the system: Passive and Active attacks

2.2.1 Passive Attacks

Passive attacks entail monitoring and listening to the communication channel of the network where security requirement privacy gets compromised [31]. Some of the passive attacks are the following:

- **Monitor and eavesdropping** Adversaries secretly listens to the communication channel to discover the contents and the behaviour of the network and its configuration. The adversaries mainly go for the communication links present near the edge location so that they can listen to aggregated data coming from the source and intermediate sensors.

- **Traffic analysis** Since the encrypted packets can provide confidentiality, adversaries can still watch the communication patterns and follow the actions involved. Adversaries mainly look for critical nodes of the network to cause more harm to the sensor network.

- **Camouflage adversaries** The adversaries mask their identities and secretly try to get involved within the network acting as a legitimate node. It is a kind of

passive attack as the infiltrated node can act as a passive listener thereby intercepting the messages within the network. Among other activities, these nodes reroute traffic and misdirect the control packets while recording information about the network's condition.

2.2.2 Active Attacks

Active attacks refer to an unauthorised way of manipulating the data while forwarding or at the aggregation stage. These can be subcategorised depending upon different network layers as explained below:

Physical Layer

- **Jamming Attack** In this Denial of Service (DoS) attack, the attacker may install a jamming source near a sensor node that would interfere with its radio frequencies affecting data transmission and block a portion of the network from communicating with other nodes.

- **Physical Attack** It is also referred to as a tampering attack where the main purpose is to access the hardware apparatus and gain full control using direct physical access in addition to cryptographic keys. An adversary can also modify the functioning of a node by reprogramming or derive secret information from shared nodes [23].

- **False Data Injection Attack** This involves an act of tampering where inaccurate sensor measurements are provided to the base station. When the attackers find it hard to tamper with the node, they mainly manipulate the sensed environment or induce false readings during data forwarding. With the help of compromised nodes, the adversary fulfils the purpose of distorting values, trigger false alarms or mask the actual event eventually compromising the integrity of the whole system [15].

Data link layer

- **Exhaustion Attack** The adversary exploits the functionality of the system, which requires control packets for communication between the nodes. Attacker

continuously sends useless data e.g. RTS (Request-To-Send) control packets and forces another node to stay awake after receiving a reply and aims for more consumption of energy resources [5].

- **Collision Attack** In this attack, the attackers send their packets at the same frequency causing disruptions in the network. When both the packets collide, it causes packet mismatch at the receiving resulting in the discarding of packets or re-transmissions. This attack can prove costly when control messages are re-transmitted, causing network nodes to lose energy rapidly, which can result in a network halt [5].

- **Collusion Attack** This is an attack where neighbouring malicious nodes may coordinate to launch or conduct more sophisticated attacks on the system. This attack may involve manipulating the performance of the network by altering critical factors such as indirect trust. In this attack, the compromised nodes are in some sort of agreement with an adversary where the adversary may collect confidential information from the system and disrupts data aggregation or trust evaluation systems through one or more compromised nodes. This may lead to packet misrouting and prompting false alarms in the network. The common defence method is to use an iterative filtering algorithm against collusion attacks [8].

Network layer

- **Selective Forwarding Attack** This is a type of packet drop attack where a compromised node does not cooperate in forwarding the data packets due to its limited resources or programmed by adversaries. This attack includes forwarding a significant number of messages and dropping a few. The probability of dropping the packet may vary depending on the level of attack [3].

- **Sink Hole/Black Hole Attack** The attacker intends to occupy all the traffic flow by forging the routing information by advertising the best possible route via itself to the sink node. A malicious node's objective is to drop packets passing through it, thus acting as a black hole.

- **Wormhole Attack** The main intent of this attack is to disrupt the routing process where an attacker records packets at one location in the network, channels them to another location within the network through a low-latency link or via a tunnel. Once the packet reaches the other end of the network, it is retransmitted into the network.

- **Hello Flood Attack** It can be considered as an energy-draining attack where an adversary broadcasts hello packets to its neighbours and neighbours which lie far by with higher power to establish communication links with them. This influences the far by nodes to believe that the adversary is their neighbour node. Neighbouring nodes are prompted to respond eventually and send packets through them, resulting in higher energy consumption. Malicious nodes receiving these packets either drop the packets or retransmits the Hello message to continue the attack [31].

- **Sybil Attack** This attack is achieved mainly through creating multiple fake identities of a node and placing them in multiple locations. The attackers objective is to disrupt the neighbour node functionality i.e. routing, nearby neighbour detection and further complicating the topology view [6].

Transport Layer

- **Flooding Attack** This attack involves a large number of compromised nodes flooding the network with multiple connection establishment requests for creating multiple simultaneous sessions within the network. This limits legitimate requests of normal nodes from getting processed and leads to exhaustion of their storage, processing, and energy resources.

- **Desynchronisation Attack** The attackers interrupts existing communication sessions between legitimate nodes by sending modified control flags or spoofing messages with bogus sequence numbers. This causes an increase in communication overhead and energy consumption due to retransmissions between desynchronised nodes. The attack exploits a vulnerability in radio synchronisation. Such attacks can be defeated by deploying strong authentication mechanisms [37].

Application layer

- **Path based DoS (PDoS) Attack** In PDoS attacks, the adversary injects either spurious or replays packets into the network when a packet is forwarded to the destination. The nodes along the path experience overhead causing wastage of energy and network bandwidth. This type of attack can be avoided through packet authentication and anti-replay algorithms [5].

- **Distributed DoS (DDoS) Attack** Denial of service are the most common attacks where multiple attacks from various locations try to exhaust network resources by sending unnecessary packets to prevent legitimate users from accessing services. Such an attack can damage a network not only by disrupting its functioning but diminishes a network's capability to provide a service.

2.3 Cluster Based Routing

With WSNs consisting of a few hundred to thousands of nodes, if every node starts transmitting data to the sink node can result in overloading the system and causing network congestion which will further lead to the dropping of a large number of packets. For this reason, as well as to cut down on energy consumption, WSNs deploy cluster-based routing where multiple sources are coupled together to form a cluster. Each cluster is managed by a cluster head, which collects data from its members and forwards it to the sink node. Thus, the cluster head acts as a gateway between the sensor node and the sink nodes.

The three main components involved in cluster routing are described below (see Figure 2.2):

- **Cluster members** They can also be referred to as sensor nodes which are used for monitoring a variety of events and transmit data to their cluster head. Each cluster member is associated with only one cluster head.

- **Cluster heads** These nodes act as an intermediary between cluster members and the sink node. They are mainly responsible for processing and forwarding aggregated data collected from their cluster members. Furthermore, they are responsible for the route discovery of their cluster members.

- **Sink Node** It is also referred to as a base station that manages all the nodes under its network, including gathering, processing, and analysing sensed data from the sensor nodes. It is located near the edge of the network so it can directly communicate with the external world.

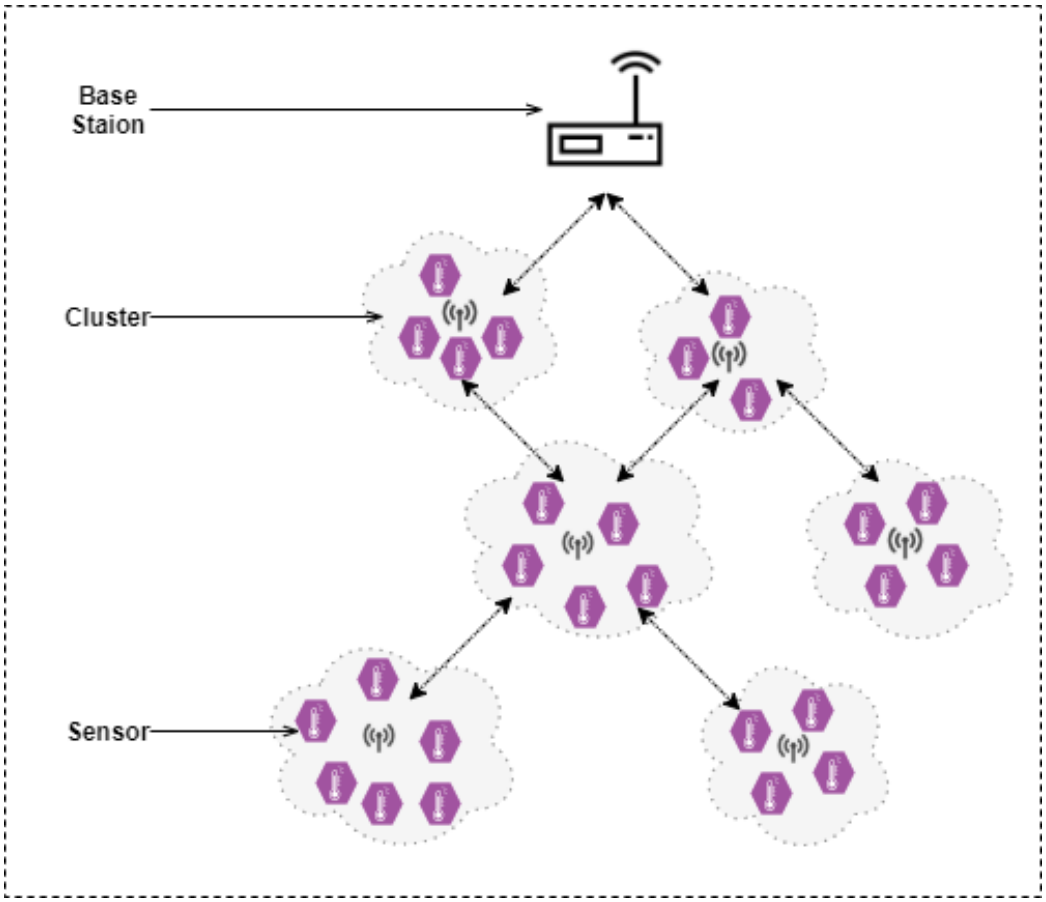


Figure 2.2: Structure of Cluster-based routing.

The cluster-based routing involves two steps, namely the clustering and routing phase.

2.3.1 Clustering Phase

Clustering is a process of partitioning large networks in sub-networks that avoids both congestion and energy consumption. This phase mainly includes cluster division, cluster head election and cluster maintenance [27].

- **Cluster head selection** Cluster heads are elected dynamically based on their

behaviour or information gathered from neighbouring cluster members. The shared information consists mainly of metrics such as energy, trust, etc. that are gathered by observing the behaviour of competing nodes.

- **Cluster formation** After the cluster head is selected, the cluster head sends join messages to its neighbours, which ultimately results in the creation of a cluster.

- **Cluster maintenance** Due to dynamic changes in network behaviour over time, cluster heads are re-elected as soon as their energy drains and they are unable to perform their functions as cluster heads.

2.3.2 Routing Phase

Routing is a process of forwarding packet from source to destination via multiple intermediate nodes in a hop-by-hop manner. The two stages involved in this phase comprises route discovery and route maintenance [7]. Figure 2.2 illustrates the structure of cluster-based routing.

- **Route Discovery** A sensor node in a cluster-based routing can propagate its data through multihop relay nodes when the area of interest is large and consists of a static sink node. It involves finding the optimal path from source to destination through cluster heads present in the network. Each cluster head node broadcasts a route request packet to its neighbours until it reaches the sink node (destination node) and waits for the sink node to respond to the route request packet to determine a route. After route initialisation, all the cluster members forward their data to the cluster head, which routes it to the sink node following the most optimal path via intermediate cluster heads. This routing phase is beneficial because all nodes do not send route requests packets rather than only potential cluster heads does the route discovery process, thereby reducing the routing overhead [16].

- **Route Maintenance** Route needs to be optimised because certain route may contain compromised nodes or nodes showing abnormal behaviour. To counter them route discovery process is re-initiated.

2.4 Challenges in WSNs

Several challenges affect the network performance of a WSN. In this section, we discuss these challenges in detail. Furthermore, we explored the security goals that need to be addressed to ensure the security mechanism follows the principles of the CIA (Confidentiality, Integrity, Availability) triad.

2.4.1 Challenges to WSN Security

Wireless sensor networks have many security issues caused due to several reasons, namely, limited resources, wireless communication and unattended environments. The threats against security are similar to that of wired traditional systems. However, the same threat mitigation techniques cannot be directly applied on wireless sensor networks [7]. The challenges described below must be considered in designing security schemes:

- **Energy constraint:** The main problem with sensor nodes is that they have limited energy. The nodes are continuously sensing the area, which requires processing and transmission of data. These processes contribute to a majority of the energy consumption of a node. As these nodes are present in an unsecured environment, they are vulnerable to many attacks. To mitigate such attacks, extra protective measures must be put in place, which is energy-consuming, and these nodes lack extra energy backup. Therefore, this becomes a requirement to optimise energy utilisation to build a power-efficient security mechanism with optimised route discovery and computationally effective cryptographic functions.

- **Hostile environment:** Sensor nodes are predominantly required to be present in an open environment with no restrictions on physical access, unlike the computer networks of traditional societies which are stored securely in a designated server room. Moreover, these nodes can be easily tampered with by an adversary physically as well as can be reprogrammed to launch an attack. Being placed in remote areas and left idle for a long time, it becomes difficult for the network administrator to detect physical tampering or attack on the node.

- **Wireless connectivity:** Sensor nodes communicate to their neighbour nodes through a wireless medium where sensor information exchanged is not safe as it can be eavesdropped, altered or replayed back by the attackers. This may also include spoofing of identity and launching malicious data injections attacks.

- **Scalability:** The wireless sensor network may consist of thousands of sensor nodes depending on the type of application. Sensor nodes support self-configuration, but networking between nodes within the dynamic environment can prove difficult as the network size increases. The solution to this problem requires designing a highly scalable security mechanism and at the same time provides reliable values for trust for all the nodes. Managing this can be challenging because of the proximity of the nodes that may interfere with packet forwarding rates [31].

- **Unreliable communication:** The sensors operate in a lossy radio medium that is highly prone to unreliability. In order to facilitate communication between the source and sink nodes, the network and its links must be stable to ensure safe communication. Even though these packets follow multihop communication, these links are often unreliable which leads to more problems. For example, if a source node sends a packet via intermediate nodes, the packet can experience several delays or can even be dropped. Significant packet losses can adversely affect the operation of a critical system.

2.4.2 Security Requirements for WSN

The primary security goals that should be introduced to enhance the functionality of the security mechanism to provide security services are as follows:

- **Data confidentiality:** It is the most notable and crucial security requirement in the wireless sensor network. This service ensures that the data cannot be accessed in an unauthorised manner and is mainly protected using cryptographic functions involving encryption and decryption through keys. This security goal is most important for military applications which require their data to be tampered proof and received through a secure channel. Any failure to achieve this would risk users privacy.

- **Data availability:** Data availability is a fundamental security service for data accessibility to authorised users. To achieve data availability, data and resources are replicated over different places. In this way, sensitive data are protected in the case of failures caused by attacks such as the Denial of Service attack.

- **Data integrity:** It is one of the critical security goals which is necessary for decision making. It enables the user to assure that the data received at the destination is free from any kind of alteration, insertion or deletion which can be checked through various check mechanism. Some efficient technique includes digital signatures for verification of the sender's identity [24].

- **Data Freshness:** Data freshness ensures the data is up-to-date and previous packets have not been replayed into the network. WSN protects data freshness with time-based counters or sequence numbers embedded into the packets that mitigate packet replay attacks by adversaries.

- **Non-repudiation:** This service assures that a user cannot deny its participation in the communication. It is attained by digital signature where sender provides its proof of delivery while the recipient has sender's proof of identity, which proves that the message received was sent by original sender [36].

- **Authentication:** This is one of the core principles of security that help establish and validate the user's identity. As a security principle, this mechanism enables secure access control capabilities that are auditable to ensure that only authorized users can access resources thereby preventing any identities from being impersonated.

2.5 Blockchain

Blockchain is a Distributed Ledger Technology (DLT), which is essentially a distributed database that is shared among interconnected and autonomous peers. The database is made up of a set of records or transactions containing information that is combined to form a block. This block consists of various components such as reference to previous block cryptographic hash, timestamp, and transactions. The links between the blocks provide security and prevent alteration in the block data.

Therefore, once the data is recorded, it cannot be changed, manipulating the data in one block would require changing data in subsequent blocks as well. All the records represent a state change and each change is validated by peers before it constitutes to form a block [13]. Thus, a blockchain is based on chaining or concatenation of blocks as shown in Figure 2.3 where an initial block of the blockchain also known as genesis block makes the foundation for the blockchain system.

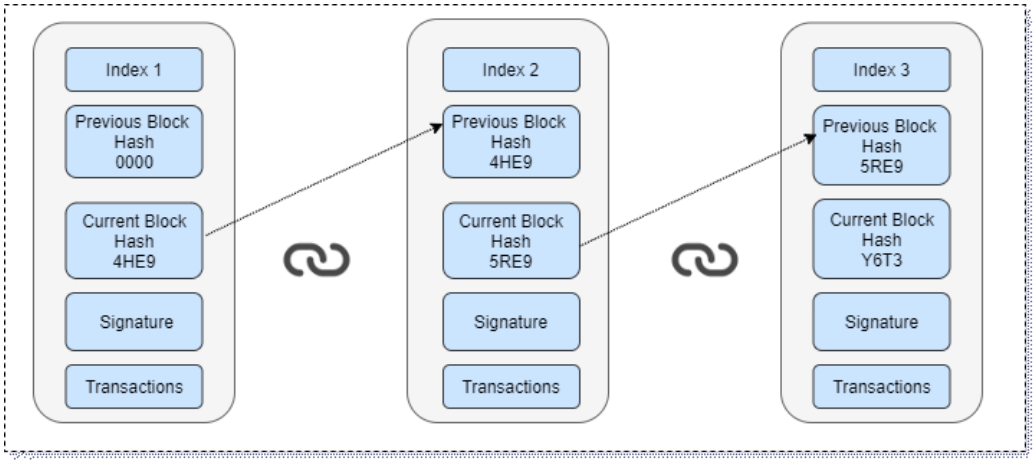


Figure 2.3: Structure of basic blockchain

The immutable ledger can also be used for tracking orders, payments, accounts and storing transparent information for all the members and brings transparency for its users. Blockchain technology was first implemented by Satoshi Nakamoto in 2008 based on an application named bitcoin cryptocurrency [22]. This permissionless digital currency works on the concept of transactions that contains information about payments between users in a Peer-to-Peer (P2P) interaction model without the need for a centralized authority. These transactions are verified through other peers based on unspent transaction outputs (UTXOs) or previous references to the asset involved in the transaction. The list of transactions is collected together to form a block consisting of backreferences to the previous block hash making it difficult for anyone to alter the previous blocks. The ownership of transactions is controlled by the use of public-key cryptography in which the transaction output is referenced towards recipients public key and the whole transaction is signed with asset owners public key [13].

There are several steps involved in processing the transactions which are as follows:

Step 1: When a new transaction is created including all the information like source id, network data etc.

Step 2: It is then entered into the pool of transactions maintained between peer to peer network.

Step 3: The peers combines transactions and perform the computation to solve a complex equation to confirm the validity of the transaction and generate a block.

Step 4: The block generated is verified by other peers and attached to the chain of blocks.

Step 5: Finally, this process confirms the transaction is complete [11].

As mentioned earlier, blockchain is a decentralised database. The features involved in this decentralised architecture are as follows.

- **Distributed architecture:** This is the most basic characteristic of blockchain in that the data is not held centrally. This allows user to access transparent data from any block present in the blockchain without worrying about its validity and vulnerability of data.

- **Anonymity:** The blockchain or state-transition system preserves the identity of interacting peer and has given ability to dynamically join or leave the peer-to-peer network. Since the data records are immutable, joining or leaving of any peer won't affect the data integrity [13].

- **Stability:** All the blocks are maintained and stored on the P2P network storage plane. This ensures that each peer has the same data view and any data change in one block needs to be updated in the rest of the blocks as well. Thus, data is stable and it is difficult to manipulate the data across the blockchain.

- **Smart contracts:** It is a sequence of instructions that are automatically executed after certain policies or rules are fulfilled. Smart contracts improve efficiency by updating the ledger through automated transactions.

2.5.1 Consensus algorithms

Consensus is an agreement between all the validators or miners to remain on the same network state in a distributed manner. The following is a summary of consensus algorithms used in blockchain.

- **Proof of Work (PoW):** The PoW is one the earliest algorithm on which today's bitcoin is built. It is based on the principle that the more you mine, the more you are rewarded. The nodes or miners are given a certain task like solving a puzzle. Once they solve and receive a consensus from all nodes the block becomes a part of the blockchain. This algorithm is computationally expensive and power-consuming.

- **Proof of Authority (PoA):** The PoA is one of the simplest consensus algorithms in the blockchain. This algorithm works on the principle that only a few designated nodes are allowed to validate a transaction and these validating nodes are capable of generating new blocks that can be part of the blockchain. Since this algorithm uses few validating nodes it consumes less computational power, more scalable and allows fast block generation. The PoA algorithm is mainly used by private blockchain enterprises [17].

- **Proof of Stake (PoS):** PoS was introduced to overcome the challenges of the PoW consensus algorithm. In PoW, all miners are competing to mine a block and the one who solves the puzzle will be rewarded whereas others are not rewarded. The PoS algorithm is based on the concept that one node with more stake or value has more mining power. The stake or value can be in form of cryptocurrency. In this algorithm, the nodes mine a block depending on its stake contribution, e.g. If a node holds 3 percent bitcoins then it can mine only 3 percent of the block. Once their transaction is validated the node gets rewarded in form of bitcoin [1].

2.5.2 Types of Blockchain

- **Public Blockchain** The public blockchain is open to everyone and does not need any approval for interacting with the network. These are truly decentralised blockchains that are immutable, transparent and make use of cryptography to enhance security. Since there are no restrictions on new nodes and to provide better security and trust among users, it requires validators to perform validation before adding entries in the blockchain. One of the first kind of permissionless blockchain is bitcoin which requires entries to include proof of work.

While certain drawbacks of public blockchain are slow transaction speeds, high power consumption and low scalabilty [20].

- **Consortium Blockchain** Private organisations make use of these types of blockchain which guarantees some degree of decentralisation where verified and authorised participants are allowed to join and granted permissions to perform activities such as participation in consensus. Certain characteristics like energy efficiency, scalability, privacy and anonymity make them suitable for businesses requiring control of the activities of participants. Ripple makes use of permissioned blockchain [28].

- **Private Blockchain** A private blockchain allows only the selected entry of users. They cannot join the network unless invited by the network administrator. This blockchain acts as a distributed ledger rather than decentralised where all the permissions and controls are restricted as well as defined by the network operator [28].

2.5.3 Working of PoA consensus

The Proof of Authority consensus mechanism is developed for permissioned blockchain having pre-authenticated validators. To provide a high transaction rate, the PoA consensus mechanism is used where we don't require high computational resources. PoA mechanism works on the concept of identity as a stake where invalid block generation by validating node will result in exclusion from the list of validating nodes. The PoA relies on sequential block generation by validating nodes at given time intervals. Each validating node produce blocks in their time interval and at the end of that interval, the next validator will start generating the block. The block generation process requires all valid transactions to be coupled in a block and the block is created which is disseminated to other validating nodes for validation and to reach a common consensus. These characteristics of the blockchain consensus mechanism make it suitable for WSNs as they operate on low power and possesses less computational resources [2]. Some other benefits of blockchain are:

- **Security:** All the data before storing it into the blockchain is validated and verified by other miners. After reaching a consensus a decision is made. This

helps in reducing risk and dependency on centralized authority for approval.

- **Trust:** With a permissioned blockchain, only authorized nodes are allowed to access the ledger. This allows in maintaining the confidentiality of data among members.

2.6 Conclusion

In this chapter, we provided the theoretical background regarding the security of WSNs. We briefly described the security attacks and discussed the challenges faced while designing WSN security. We presented some of the principles involved in routing large networks and described the components of a blockchain. Additionally, we outlined the essential security goals for maintaining the security of WSN and the blockchain's role in achieving these goals.

Chapter 3

Related Work

3.1 Trust Based Security Mechanisms

Prathap *et al.* [32] has implemented a trust-based malicious node detection scheme (CMNTS) that targets nodes that are involved in WSN attacks such as packet modification, packet dropping, identity exploitation attack and packet misrouting. The objective of this paper is to find the malicious nodes that are present in the routing path that might be involved in any kind of attacks mentioned above. The solution includes cryptographic techniques and trust evaluation for achieving the security objective. The trust is evaluated by observing next node behaviour for successful and unsuccessful interactions. The cryptographic techniques involve the use of pairwise keys to encrypt the data. This model works by adding an encrypted tag containing identity and next level node trust during data forwarding. After receiving data, the sink node starts decrypting and processes data packet to detect malicious node. The next step is to re-select the parent node for topology maintenance to keep the malicious node isolated from the network topology.

The operation of the CMNTS model begins with the data transmission where every intermediate node on the routing path appends its identity and trust on its next hop with the data packet that has to be forwarded. The information exchanged is in an encrypted format to prevent the packet from modification. Next, after the packet is received at the sink node, the packet is processed by decrypting the tags and messages. Here the tags are decrypted with the child node's key and if the tag is decrypted by any first-level child node of the sink then information of that tag is stored and the rest of the packet is decrypted by subsequent child nodes key present at the second level. The detection phase starts by evaluating the average trust of all the parent and child node at the sink and the threshold-based detection is performed on the trust values to detect a malicious node.

This technique is capable of detecting various kind of attacks, some of them are

explained as follows:

- The Sybil attack is detected when the decryption key of the node does not match with the node id in the packet.
- A packet modification attack is identified when the packet decryption process fails.
- The packet drop is detected by the child node by keeping the track of the number of packets forwarded by the parent node. The child node reduces the trust when the packet drop rate of the parent node increases. As the trust goes below a threshold the sink node can detect the corresponding node as a malicious one.
- The packet misrouting attack is identified based on the tree topology that is used by the author.

Though their model improved the detection rate, the author didn't clearly mention how they calculate the trust. Also, packet processing at the sink node is very energy consuming since decrypting tag is checked with every child node keys.

BTEM [26] introduces a Belief based Trust Evaluation Mechanism for isolation and detection of malicious nodes that use Bayesian estimation for indirect and direct trust. The author also describes securing the routing path to enhance end to end integrity by detecting the malicious nodes involved in attacks. They tried to incorporate a trust mechanism by defining trust as a belief of nodes on each other and computing trust based on behavioural interactions.

Their model is divided into three modules, namely, the traffic monitoring module, trust evaluation module and decision-making module. The traffic monitoring module observes the packet forwarding behaviour of neighbouring nodes. The trust evaluation module estimates trust based on direct and indirect evaluations of past interactions. This module is further divided into three sub-modules that calculates packet send, packet receive and packet in transit. With the packet information from these sub-modules, it is possible to evaluate the direct trustworthiness of the node by looking at the ratio of packet forwarded factors at different time intervals. The indirect trust uses the Bayesian theorem to weigh the node's trustworthiness. Furthermore, it uses the Decision making module to determine if the node conduct is malicious or normal

by checking its trust against the threshold. If the module detects any malicious nodes, it isolates the corresponding node from the network.

This model improved data communication reliability by minimising the internal threats caused by DOS attacks, Bad mouth and on-off attacks. However, it only considered packet forwarding behaviour as the only metric for trust calculation. Since the packet forwarding behaviour is not the only single parameter for evaluating trust, the author didn't focus on using other parameters such as energy consumption.

In [19], a hybrid trust Intrusion Detection System is proposed for clustered WSNs based on the trust model and misuse based detection. Their solution proposes the sensor nodes periodically exchange control packets with base station containing their neighbour nodes trust values which are evaluated based on direct and indirect evaluations. Further, the base station decides on nodes misbehaviour and avoids the packet routing through misbehaving nodes. The trust evaluation is based on identifying and observing five behavioural activities like reliability in communication, sensed data, etc. Their model detects different types of attacks and increases network lifetime by preventing malicious node behaviour. Even though the model claims to detect several attacks, it fails to provide any significant results in terms of model detection rate and accuracy.

Yuxin Sun *et al.* [30] provided an improvement in the trust management model where reputation and threshold for malicious detection are dynamically computed. A beta distribution function is used to derive trust from the communication behaviours of sensor nodes. The model helps in resisting several internal attacks such as selective forwarding, on-off and Bad mouth attacks. They achieved a better detection rate with a low false alarm rate by adopting a reputation threshold based on the average threshold level of two clusters i.e. normal and malicious node. The limitation of the work is that the author assumes only two cluster centers however, realistically one can assign more than two clusters centers leading to the spreading of malicious nodes in normal clusters that can go undetected and cause problems in the long run.

Ahmed Saidi *et al.* [27] introduced a trust management scheme for secure cluster head (CH) election and its misbehaviour detection depending upon three trust types such as data, communication and energy. They have also considered the scenario of compromised CH after an election where trust evaluation is performed at both base

station and cluster member level to find malicious behaviour of CH. Once malicious CH is detected, they adopt a local clustering algorithm to isolate the CH and assign a new CH to the affected cluster members. The proposed scheme not only prevents malicious nodes from becoming CHs but also isolates the compromised CH after the election with fewer false positive and negative alarm rates. However, the author did not focus on the historical values of trust for trust evaluation.

Hierarchical Trust Management System (HTMS) is proposed by Alexander Basan *et al.* [4] for securing the network from internal attacks. The main contribution of this paper is to find malicious nodes based on direct communication trust and centralised trust value calculated at a higher hierarchical level comparing the load and residual energy of different cluster members. Determining trust value at a higher level rather than at the same level of the network enabled to prevent more usage of power resources and network bandwidth. The shortcoming of this paper is that the system will fail when the cluster head is attacked resulting in manipulation of the trust values for their cluster members.

3.2 Blockchain-Based Security Mechanisms

In SenseChain [9], the author presented a distributed anomaly detection system for identifying false sensors and utilising blockchain for recording anomaly behaviour. They examined a scenario during the post-sensing phase where the sensors employed in a hostile environment can report incorrect or biased reports resulting in wrong decisions and false intermixing of information. In the solution, they proposed to distinguish between good and bad nodes based on an anomaly detection algorithm that assigns a reputation to each node, which is then used in weighted aggregation algorithms.

The solution provides all sensor nodes to broadcast their sensing reports to validators for peer validation. The report contains information about the Signal to Noise Ratio (SNR) and distance to the target that the validator uses to determine the target's precise location and its validation zone. Sensor nodes outside the validation zone are considered malicious, while those inside the validation zone are regarded as normal and each normal node is assigned a confidence score on their truthfulness on the information exchanged.

In the following step, the confidence score is recorded in the blockchain employing a transaction, and the block is generated with heterogeneous difficulty assigned to that block's validators for their credibility, thus increasing the competition among themselves. They have also designed a consensus mechanism called as Most-Difficult-Chain consensus where all validators choose the block with the highest difficulty in terms of PoW.

Next, the confidence scores from all blocks to the genesis block are employed to determine provenance and the current reputation of the sensor node by defining a non-linear function using historical confidence scores and difficulty associated with each block. Finally, these reputations are used by validators to perform weighted aggregation of sensing data for detecting and locating a target. This model provided a tamper-proof means to arrive at a distributed consensus among trustless entities but uses energy-consuming Proof of Work consensus algorithm.

Wei She *et al.* [35] performed malicious node detection using blockchain-based trust model and smart contracts. The author outlines the model where all the communication data is recorded in blockchain data structures, which then serve as the basis for assessing the node's credibility using three parameters including processing delay, forwarding rate, and response time. The solution assigns a score to each node based on its credibility, this score determines the degree of the node's maliciousness. The blockchain validators use a voting consensus mechanism to achieve a common perspective on nodes maliciousness. Further, it uses normal nodes for locating the unknown nodes present in the network using the quadrilateral measurement localization method. In addition, the model demonstrates effectiveness through a consensus process, but the paper's shortcoming is the consensus method's energy consumption along with its high computational power consumption requirements.

Sarah Asiri *et al.* [6] introduced a blockchain-based architecture for trust modelling of IoT devices. The proposed model utilizes hyperledger fabric blockchain and smart contracts to resist Sybil attacks and analyse the trustworthiness of devices before actual communication starts between participants or devices. This model implements identity management and authenticates that transactions proposed are coming from verified and trusted sources while maintaining the integrity of messages. The author primarily focuses on the Sybil attack and uses only the packet delivery rate for trust

evaluation.

In [33], Volkan *et al.* explained the layered architecture of the blockchain-based trust mechanism for IoT that evaluates the data trustworthiness and its transaction verification at the blockchain layer. The trust in observational data is augmented by using node reputation, confidence in its data correctness and correlation among neighbours data. They also implemented the custom private-blockchain model consisting of periodic interval-based block generation, reputation-based block validation and distributed consensus mechanism. Their approach for generating blocks sequentially by different validators have improved the overall performance. The paper primarily focuses on data-based attacks and does not discuss how their blockchain model can be extended to detect other attacks.

A Blockchain-based routing scheme is proposed in [17] by Jidian *et al.* for enhancing the trustworthiness of routing information between routing nodes in wireless sensor networks. Among the key contributions of this work is the development of a routing information management system based on blockchain token transactions and the determination of the next optimal routing node through reinforcement learning, where tokens represent the digitised information of a packet. The learning model at every routing node gathers information from the blockchain network and returns a routing policy to help choose the best optimal path. This enables the learning model to find the best route rather than relying on neighbouring nodes, thus avoiding the malicious nodes in the path. In the blockchain, the server nodes manage the contracts and verify the transactions, while routing nodes interact and initiate the token contracts to map packet state information and generate tokens. The server nodes act as validators and use the Proof of Authority (PoA) consensus algorithm to reach consensus without using a computationally expensive mining process. However, each routing node has to communicate with the blockchain to confirm the packet received that can cause delay. This delay can add additional time overhead causing the malicious node to remain undetected for a long time which can pose a security threat.

3.3 Summary

In summary, we can observe that most of the papers include either trust-based or blockchain-based security mechanisms summarised in Table 3.1. In some cases, both

Table 3.1: A comparison of features for WSN security based works.

Approach	**Trust Based**	**Blockchain Based**	**Attacks Defended**
CMNTS [32]	✓	×	Packet Modification & drop, Sybil, Bad mouth
H-IDS [19]	✓	×	False Data, Packet drop
BTM [35]	✓	✓	Packet drop
SRTM [6]	✓	✓	Sybil,Replay, Bad mouth
BDTM [30]	✓	×	Packet drop, Bad mouth, On-off
BTA [33]	✓	✓	False Data, Impersonation
SenseChain [9]	✓	✓	False Data
BTEM [26]	✓	×	False Data, On-off, Bad mouth
CH-TM [27]	✓	×	False Data, Packet drop
HTMS [4]	✓	×	Flooding, Sybil, Blackhole
RLBC [17]	✓	✓	Packet drop, Blackhole

techniques have been used, however, most papers that use blockchain-based mechanisms rely on the PoW consensus mechanism that is extremely resource-intensive for low battery devices such as wireless sensors. It is also notable that some papers using blockchain-based techniques only use blockchain as a storage mechanism while there is no effort to extend the scope of blockchain usage. Therefore in this work, we have designed a new framework that uses both the techniques and the PoA consensus mechanism. We also showed through our evaluation how the proposed work can provide an improvement over some existing models. In our next chapter, we will discuss our design and research methodology, including our framework's algorithm for reputation assignment and detection of malicious nodes.

Chapter 4

Design and Methodology

4.1 Research Methodology

In this chapter, we first outline the problem definition and the component models used in the framework. We then describe the design of the blockchain-based reputation system for securing WSNs. Finally, the functioning of the proposed framework for malicious node detection and blockchain management is discussed.

4.2 Problem Definition

In this thesis, we address the issue of malicious nodes present in WSNs disrupting routing activities. The objective is to identify these malicious nodes and minimise the likelihood of selecting them as a next-hop node by prohibiting them from routing activities. Malicious node detection is a crucial task as it poses a security challenge to several applications that requires a high level of confidentiality, integrity and authenticity. Different approaches are proposed to detect and prevent malicious activity as was discussed in Chapter 3.

We propose a solution by developing a framework that has two parametric components namely trust and reputation. We also develop an algorithm that calculates the confidence score using the energy consumption of nodes and use blockchain to store and validate the information of the data packets. The framework also calculates the reputation based on historical confidence values. Furthermore, we detect the malicious nodes by setting up the threshold for trust and reputation. Based on our survey, we find that there has been no effort to solve the malicious node detection using a proof of authority consensus algorithm. This is the research gap addressed in this thesis. In the next sections, we will define our research methodology, algorithms, and working model.

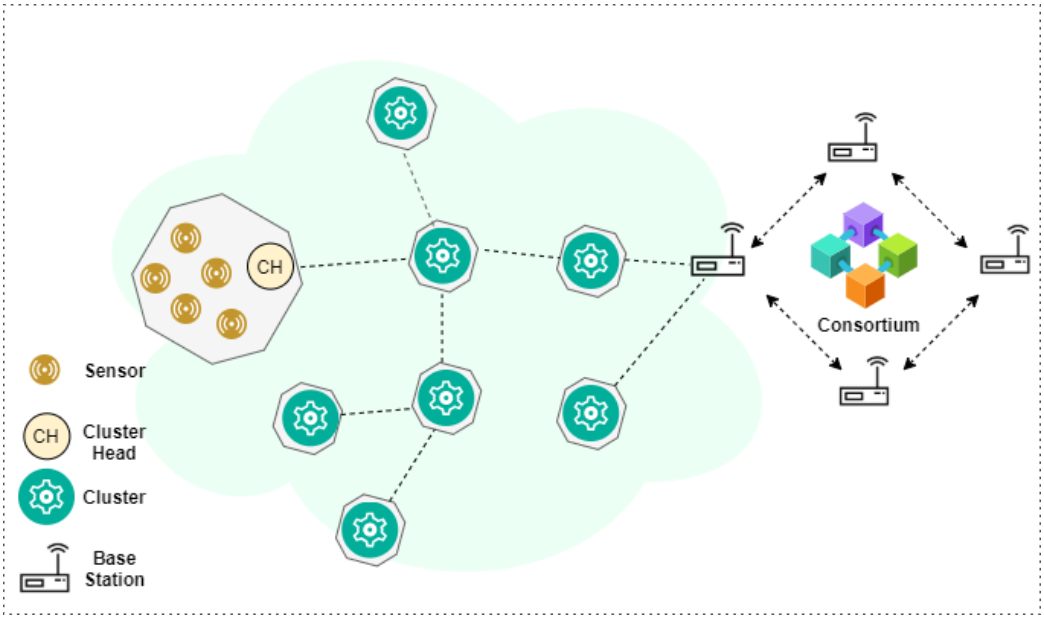


Figure 4.1: Network Setup for Proposed Work

4.3 Components

4.3.1 System Model and Assumptions

Figure 4.1 shows the architecture for the WSN topology considered for our proposed framework. The proposed model consists of cluster-based WSN with blockchain consortium and static sensor nodes deployed in the sensing environment. The network is categorized into three tiers: sensors, Cluster Heads (CHs), and Base Stations (BSs).

- **Sensor** The sensors are used to detect an event from the sensing area and transmits sensed data to the corresponding cluster head. The sensor belongs to one cluster network and cannot perform complex operations because of limited energy and computation capacity.

- **Cluster head** These nodes are considered to have higher energy, computational power and transmission range. It is responsible for gathering data from sensors, performing data aggregation on the received data and then route processed data to the base station through other cluster head nodes. The cluster head evaluates their next-hop neighbour node trust value through a watchdog mechanism and shares it with the base station via multi-hop routing. While CHs can carry

out encryption where intermediate CHs encrypts their ID, energy and next-hop trust value and append it with the data packet that has to be forwarded.

- **Sink node/Base station** It manages all sensor nodes in the network, including data analysis, assigning identities and pre-shared keys, recording sensors trust values and detecting an attack [19]. With higher memory and processing capabilities, the base station acts as a trusted entity and gateway for blockchain. It carries out functionalities like managing blockchain and transaction data.

For our study, it is assumed that the base station is not compromised. A sensor node is considered malicious if it has been compromised by a malicious attacker who may manipulate the data packets received from other CH devices. Finally, we assume that all the sensor nodes are in an active state and they are trustworthy and not compromised if they forwards all the packets to the next node.

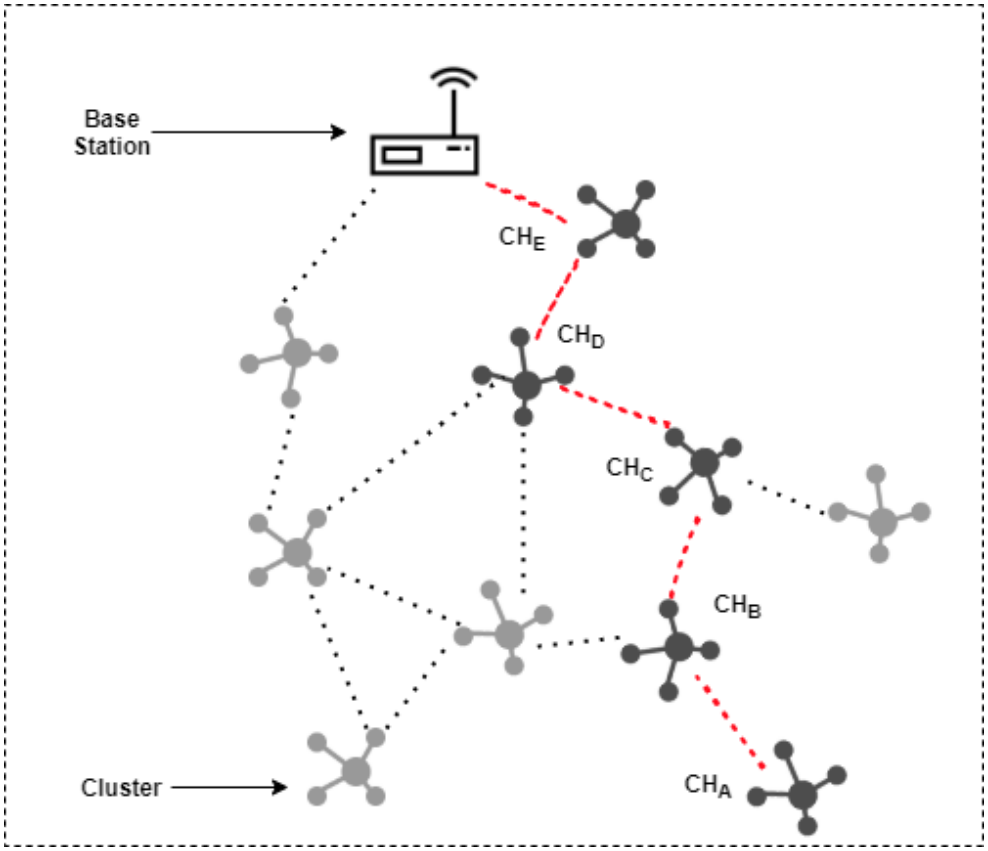


Figure 4.2: Traffic flow in clustered WSN

Figure 4.2 illustrates the cluster-based WSN topology, where CH_i represents the

cluster head that communicates with their cluster members and forwards their data to the next-hop cluster head node, also referred to as the parent node of CH_i, while the sender is known as the child node. For e.g., in Figure 4.2, we see that CH_A forwards packets to the parent node CH_B. This sequence is repeated by CH_B and the subsequent cluster heads until the packet reaches the base station. This transmission shows how the packet is transmitted to the base station via intermediate cluster head nodes.

4.3.2 Threat Model

Wireless sensor nodes are susceptible to many attacks and are equally vulnerable at each tier of the system model. Generally, the attack does not occur during the network initialisation phase and even if it occurs our framework won't be able to detect the attack. However, the attacks can occur during the routing phase when clusters heads collect and transmit data to the sink or other cluster heads.

These attacks can be divided into two categories, namely, internal and external attacks. Internal attacks are where an attacker can launch the attack within the network by programming compromised nodes. These attacks include selective forwarding, Sybil, packet modification and replay attacks, whereas node insertion attacks and eavesdropping attacks are considered as external attacks where the adversary attacks from outside the network. We now focus on the attacks considered in this thesis.

In a **node insertion attack**, a fake node acting as a legitimate node is inserted in the network that engages in routing activities. These nodes may misroute the data packet or it may drop packets received from other nodes. In contrast, an **eavesdropping attack** is a passive attack where an attacker intercepts the packet through packet sniffing tools. This allows them to reveal crucial information about the network.

Next, the **selective forwarding attack** enables the compromised node to be programmed by an adversary which drops the packet selectively and affects the network performance. Also, the node parameters may fail to reach the sink node preventing topology maintenance.

The next is the **Sybil attack** which is responsible for creating duplicate identities of the node that will redirect the traffic to the compromised node instead of going

to the legitimate node. The compromised node can affect the data transmission and reveal confidential information to an attacker.

Finally, the attackers can cause **replay attacks** by retransmitting the earlier messages back into the network, thus causing traffic congestion and difficulty in handling the traffic.

4.3.3 Blockchain Model

By employing blockchain technology, the framework design uses a distributed and decentralised identity and trust management system, thereby avoiding the risks of a single point of failure. Blockchain enhances security and end to end data reliability by assuring the data is coming through uncompromised nodes. This also provides authentication of nodes and protects the network from various kinds of threats. There are several parameters involved with the blockchain model that are mentioned as follows:

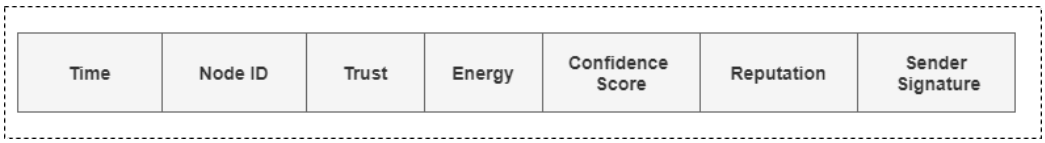


Figure 4.3: Sample transaction

- **Transaction** It is a record that stores information of individual sensor nodes containing trust, energy level, confidence scores and reputation. Figure 4.3 shows the sample record of a transaction for our framework.

- **Block and Blockchain** Across the blockchain, the transactions are aggregated by validators to form a block. The new blocks are joined with the existing chain after all validators reach a consensus. Each block is made up of two components: firstly a block header which contains the hash derived from the hash of the previous block and the Merkle tree hash of the transactions. This ensures the integrity of transactions in the blockchain. Secondly, the block body contains the set of transactions used for auditability purposes. Figure 4.4 represents the designed block header and the block body consisting of different transactions (TXN1, TXN2 etc.) and their corresponding hashes (Hash1, Hash2 etc.). The

blockchain is employed to determine the reputation of each sensor by extracting historical confidence scores from it.

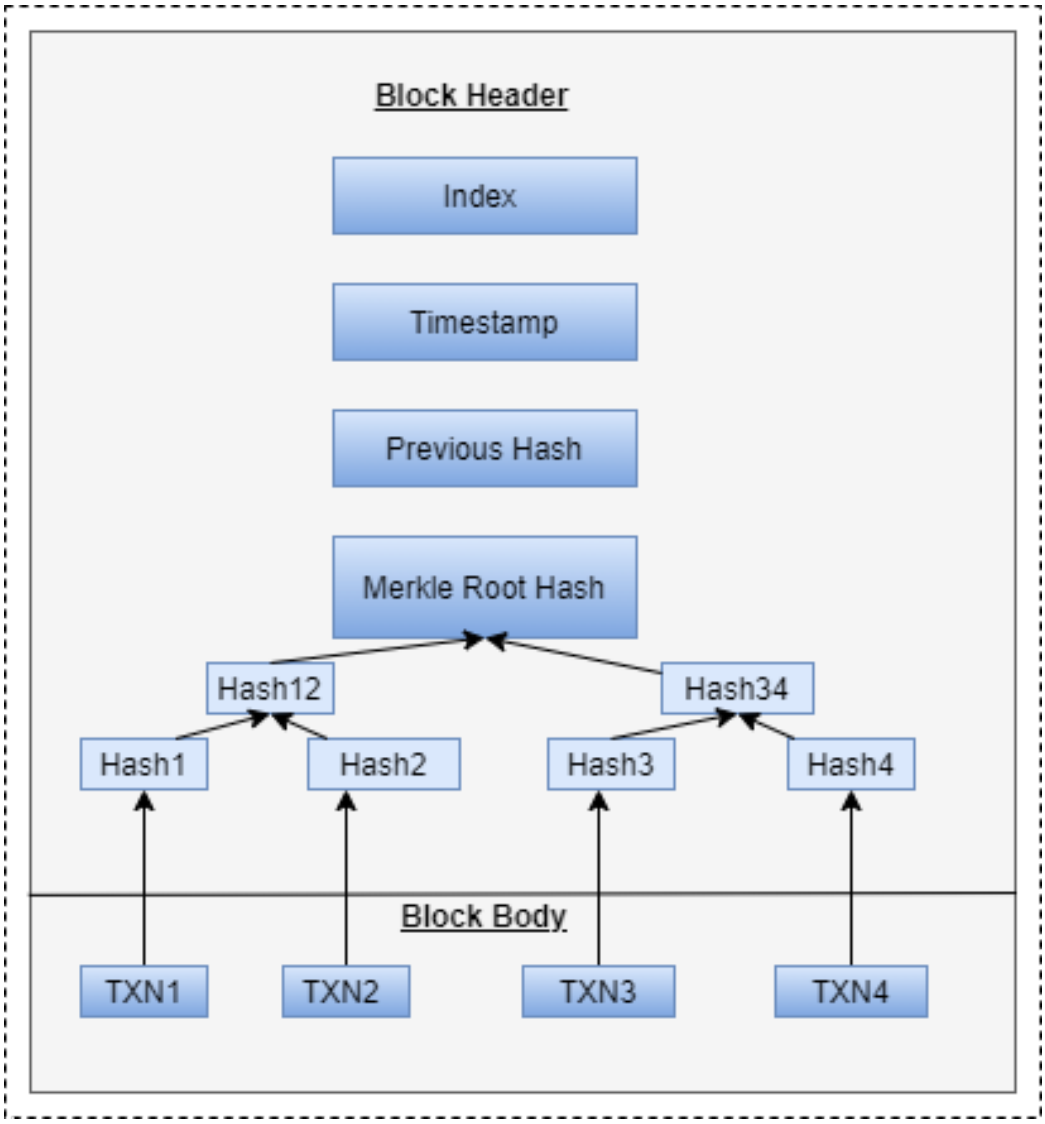


Figure 4.4: Standard block structure

Node Enrollment and Key Allocation Before deploying sensor nodes into the environment, they are registered first under the sink node to obtain credentials. They have a unique identity and pair of public and private keys generated by a trusted Certified Authority (CA). The blockchain CA also provides them with enrollment

certificates that act as a digital signature that ensures the nodes are uniquely authenticated. The registration list for these nodes is retained in the blockchain, thus allowing authorized devices to join the wireless sensor network.

4.4 Proposed Scheme

4.4.1 Trust Score Calculation

In this section, we will discuss the trust calculation mechanism in wireless sensor networks. Packet forwarding is an important step to ensure that the packet reaches its destination, but an adversary can exploit this by attacking a node and configuring it to drop packets selectively. To address this issue, we use communication trust as a parameter to identify whether the packets are forwarded successfully or dropped. We define communication trust as a packet forwarding ratio and is measured as the ratio of packet successfully forwarded by the parent node against packets sent by the child node. Here the node observes the parent node in the promiscuous mode and overhears the communication to determine its trustworthiness. It determines trust by recording the successful forwarding of packets in each stipulated time and if it fails to forward packets in that time, then it is considered as an unsuccessful interaction and the trust value drops accordingly. We calculate the communication trust (C_T) using the expectation of beta distribution function [29] where the equation can be written as follows:

$$C_T = \frac{S + 1}{S + U + 2} \tag{4.1}$$

Let S be the number of successful interactions and U be the number of unsuccessful interactions of the node, and these interactions are classified based on the sincerity of providing packet acknowledgements by the node. Then, the child node calculates the communication trust for its parent node using the equation 4.1.

4.4.2 Blockchain-Based Reputation System

Our framework is mainly designed for detecting the malicious nodes which are participating in the routing activities. These malicious nodes show selfish behaviour by dropping some packets to save energy resources. Our reputation system uses energy as the core component for estimating the reputation of the node. The idea behind

this work is that if the node experiences too many fluctuations in its energy consumption with respect to the corresponding node level, then it may be involved in some malicious activity. The system involves two phases:

Confidence score calculation:

The working principle behind the calculation of confidence scores is that the energy consumption of nodes at the same level is correlated. The self-assurance or confidence level of the node on its energy usage with respect to nearby neighbour's energy usage is termed as confidence score as shown in Equation 4.2. These scores are recorded on the blockchain with other information such as trust and energy level as a transaction. For computing the confidence score, we assume that the load is balanced, i.e. energy consumption for an activity would be the same across all nodes of the same level. Also, the neighbour node used for comparison should be uncompromised and perform the same operation. From Equation 4.2, we can see that the neighbour nodes energy consumption is inversely proportional to the confidence score of the node whose value is to be calculated. So, if the malicious node saves energy by sending fewer packets, it has a low energy consumption compared to its neighbour node. On the other hand, if the neighbour nodes energy consumption is relatively higher than the malicious node, this will lead to assigning a low confidence score for the malicious node.

$$CS_{mn} = \frac{EC_m}{EC_n} \tag{4.2}$$

Here CS_{mn} is the confidence score for node m relative to neighbour node n, EC_m is the energy consumption of node m and EC_n is the energy consumption of neighbour node n.

For enhancing our framework, we choose to use another parameter called *reputation* which is explained in the next section.

Reputation Estimation

A nodes historical reputation indicates the degree of its reliability and behaviour over time. During the estimation process, the node's long term confidence scores are employed. These confidence scores are extracted from the genesis block to the most recent blocks with historical transactions containing the node's identity.

Equation 4.3 indicates the computation of reputation using the non-linear function.

$$R_m = e^{\log(1-\beta V_{cs})} \tag{4.3}$$

where R_m is reputation for node m, β is scaling factor, V_{cs} is variance of confidence scores.

We included variance of confidence scores in calculating reputation to get the overall idea of the spread of the confidence values. Since, variation in packet dropping rate causes nodes' energy consumption to fluctuate, resulting in shifting of confidence scores. Thus, to capture these inconsistencies in confidence values, we evaluated the variance of historical confidence scores. In the case of normal nodes, the reputation will remain constant because of fewer fluctuations in the confidence scores, whereas malicious nodes will have high fluctuations that will spread confidence scores more widely, ultimately the overall reputation for the malicious nodes will decrease over a period. When it goes below a particular threshold, we declare the node is involved in some malicious activity.

We developed an algorithm for the reputation computation of a node and presented it in Algorithm 1. Our algorithm evaluates a node's reputation by calculating a confidence score based on the neighbour node and estimating its reputation. First, we extract the previous energy levels from the blockchain for sensor node S and its neighbour node N using the function $ExtractEngBC(S_{id},N_{id})$, resulting in previous energy levels for nodes S and N as PE_S and PE_N, respectively. Additionally, current energy levels are obtained for node S and N as E_S and E_N as input to the algorithm. These energy values are used to determine the energy consumption of nodes and then calculate the confidence score for a node S (line 2 to line 4). For reputation estimation, the function $ExtractCsBC(S_{id})$ allows the user to extract historical confidence scores from the blockchain and then the variance of confidence scores is computed as shown by $CS_{variance}$, while $len(CS)$ represents the number of confidence scores, CS_{sum} and CS_{mean} shows the sum and mean of confidence scores, respectively (line 5 to line 13). At last, reputation is evaluated using the equation 4.3 (line 14).

Algorithm 1 Reputation Computation Algorithm for a node

Input:

Sensor Id: S_{id}

Sensor Energy: E_s

Neighbour Id: N_{id}

Neighbour Energy: E_n

Output:

Reputation: R_s

1: $PE_s, PE_n \leftarrow$ExtractEngBC(S_{id},N_{id})
2: $EC_s \leftarrow E_s - PE_s$
3: $EC_n \leftarrow E_n - PE_n$
4: $CS_s \leftarrow EC_n / EC_n$
5: $[CS_s] \leftarrow$ExtractCsBC(S_{id})
6: **for** $i = 0$: $len(CS_s)$ **do**
7: $\quad CS_{sum} \leftarrow CS_{sum} + CS_s[i]$
8: **end for**
9: $CS_{mean} \leftarrow CS_{sum} / len(CS_s)$
10: **for** $i = 0$: $len(CS_s)$ **do**
11: $\quad SquaredDiff \leftarrow SquaredDiff + (CS_s[i] - CS_{mean})^2$
12: **end for**
13: $CS_{variance} \leftarrow SquaredDiff / len(CS_s)$
14: $R_s \leftarrow e^{log(1-\beta \times CS_{variance})}$
15: **return**R_s

4.4.3 Workflow of Proposed Framework

Once the nodes are registered with the blockchain CA, they are provided with their keys, i.e., public and private key. They are then deployed into the environment and establishes a cluster-based topology. In our work, we are not focussing on how CH selection is performed and route formation is done as it is not the scope of this thesis. Our main objective is to detect malicious nodes after data transmission, which includes evaluating the trust and reputation of individual nodes. Figure 4.5 shows the workflow of our proposed framework.

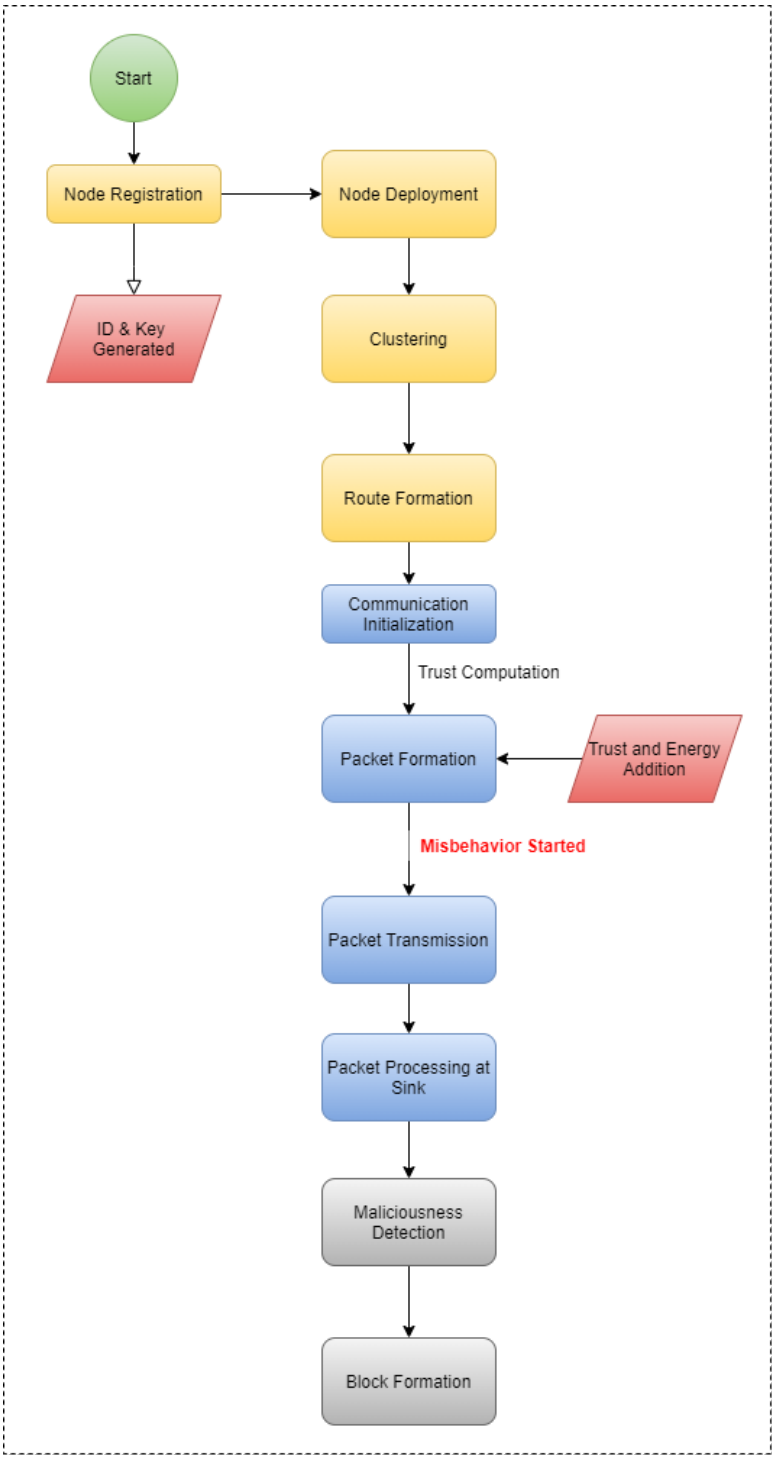


Figure 4.5: Workflow of the framework

During data transmission, we use trust and cryptographic key components. The trust component acquires the true behaviour of the node by observing its packet forwarding behaviour and determines its trustworthiness level by its packet forwarding

ratio to other nodes. Each CHs determines the trust value of its parent by observing its successful and unsuccessful transactions. The cryptographic key component, on the other hand, is responsible for securing packet against unauthorised entities. This includes packet encryption with the sink node's public key and decryption of those packets using the sink node's private key.

The malicious nodes in the computed paths are identified during the detection phase, which requires every node to append its id, residual energy and trust on its relay node to the data packets as encrypted tags and send them to the relay node with the packet. The tags are added sequentially by intermediate nodes until the packet reaches the sink node. The decryption process begins at the sink node, where all the tags are decrypted using the sink node's private key, which allows the sink node to gather all the trust values and energy levels of each intermediate node. Later, the sink node calculates the confidence score for the corresponding node and stores their data in blockchain to create a set of historical records.

During the detection phase, the sink node relies on the confidence scores and trust level of intermediate nodes to detect the benign or malignant behaviour of nodes. The sink node estimates the reputation of the intermediate node based on historical confidence scores. Also, it determines the average trustworthiness by aggregating the trust scores received from various child nodes. Both trust and reputation are verified against the predefined threshold to decide on its maliciousness.

Apart from this, there are several processes involved in the functioning of the framework. We will discuss them one by one.

Traffic Generation

Following Figure 4.2, let us consider a source CH A that has to send data of its cluster members, it creates a message and encrypts the message with the sink node public key S_{puk} to generate a packet $m_A = \{id_A, \{T_B, Eng, Sq, D\}\}$, where id_A is the ID of node A, T_B is the trust on its parent node B, Eng is the current energy level of the node, Sq is the sequence number of packet, D is data. To validate the identity of node A at the sink node, T_B, Eng, Sq and data D are encrypted by node A private key as seen in m_A. Node A forwards message m_A to next-hop parent CH node B. Once CH B receives m_A, it forms an encrypted tag $m_B = \{id_B, T_C, Eng\}$ containing node

id, trust on its parent node C, and current energy level and attaches with the packet received to form $[m_A, m_B]$. This process is followed by all subsequent intermediate nodes until the packet reaches the sink node.

Packet processing

The packet received at the sink node contains a message from the source node and a sequence of tags added by intermediate nodes in the routing path as depicted in Figure 4.6.

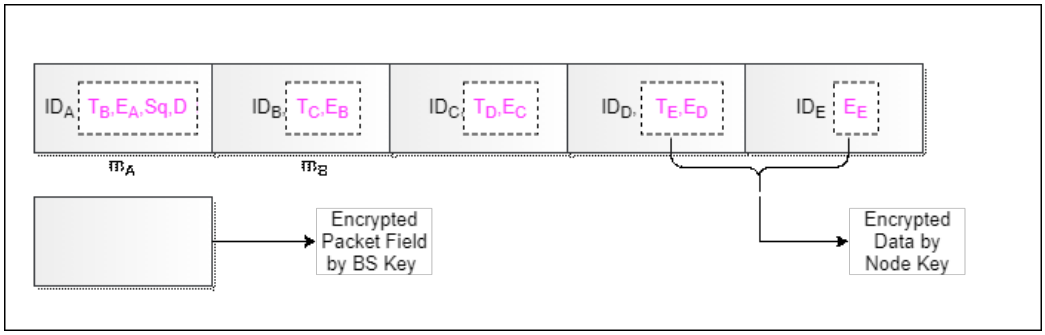


Figure 4.6: Example of data packet

Step 1: Firstly, the last tag is decrypted by using the sink node private key to generate decrypted message m' which is in the form of $\{id, \{Eng\}\}$. Next, node ID id is checked from the registration list present in the blockchain. If validated, the public key of a node with ID id is used to decrypt $\{Eng\}$ that is still encrypted in the decrypted message m'. Once decrypted, node energy information is collected. If the lookup for ID id in blockchain fails, the sink node may detect a node insertion attack and the packet is discarded entirely. Whereas, failure in decrypting the inner encrypted part of message m' will be detected as an impersonation attack.

Step 2: The next encrypted tag is removed from the packet and step 1 is performed on the rest of the packet to achieve trust and energy factors. The remaining tags and source message are decrypted orderly and any failure in decryption will lead to the identification of a packet modification attack.

Step 3: Lastly when the source message is decrypted, it will be in the form of $\{id_A, \{T_B, Eng, Sq, D\}\}$, the inner part is decrypted with the source node public key to determine data, trust, energy level and sequence number. The sequence number is verified by the sink node to detect any replay attacks.

Step 4: Finally, the sink node records all the trust and energy levels of nodes for malicious node detection.

Detection process

After data transmission, the sink has a list of trust values and confidence scores for each node. These confidence scores are used to estimate the reputation of each node. The sink node aggregates all the trust values for the parent node received from its child nodes to compute aggregated trust. If both the aggregated trust and reputation of a node is below a defined threshold for a given trust and reputation, then the node is identified as malicious. In case if any of the parameters is higher than thresholds, the node is classified as a normal node.

Algorithm 2 Malicious Node Detection Algorithm

Input:

Sensor Id: S_{id}

Trust :T_s

Trust Threshold: $T_{Threshold}$

Reputation Threshold: $R_{Threshold}$

Output:

IsMalicious: M_s

if $T_s \leq T_{Threshold}$ **then**
  $R_s \leftarrow ReputationComputation(S_{id}, E_s, N_{id}, E_n)$
  if $R_s \leq R_{Threshold}$ **then**
    $M_s \leftarrow True$
  else
    $M_s \leftarrow False$
  end if
end if
return M_s

Blockchain Process

This process is followed by updating the node information in the blockchain. All the node information is recorded in the form of transactions. Each sink node mines

the block containing transactions of its network nodes. After the block is mined, the block is broadcasted to other validators for verifying the block transactions. This may include verifying the transactions, validating the assignment of confidence scores and estimated the reputation of the nodes by their respective sink nodes. We are using a permissioned blockchain, where only sink nodes are allowed to generate the blocks based on the PoA consensus algorithm. Once a sink node starts creating a block the other sink node needs to wait for its turn to mine a block. All the validators verify the block and hence achieve a consensus. The blocks are then committed and appended to the blockchain network. If any of the validators finds an invalid transaction when the conditions are not satisfied, they send the warning message about the transaction thus prevent the block from adding to the ledger.

4.5 Summary

In this chapter, we discussed our framework for detecting malicious nodes with a detailed description of each process. We have shown our reputation management system that determines the confidence score and reputation of the nodes using the algorithm as discussed. Then, the proposed approach is presented, which comprises multiple processes such as traffic generation, packet processing and detection process. In the next chapter, evaluation methodology and security analysis are discussed.

Chapter 5

Evaluation Methodology and Analysis

This section will describe the experimental evaluation for our proposed framework. We evaluate the performance using the Network simulation tool (NS3) after performing various simulations. Additionally, we have compared our proposed framework with an existing security model BTEM. Several parameters such as detection rate and trustworthiness of nodes are studied while varying the number of malicious nodes present in the network.

We tested our framework by implementing it in a discrete-event network simulator NS3. NS3 is an open-source simulator platform to create network simulations by developing real systems in a virtual environment. All the system requirements like network topology creation, packet flow, application-specific functionalities are modelled to study the system behavior [10]. The data analysis and system visualisations are other components of this simulator.

In the next section, we will first discuss the experimental setup as well as the performance metrics used to examine the framework. Next, we will review the experimental results. Finally, the comprehensive security analysis of the framework will be analysed.

5.1 Experimental Setup

In this evaluation, the sensor nodes are deployed in the region of 100m X 100m square area with all the nodes acting as source nodes. The sensor nodes are considered static with the same initial energy. Initially, all the sensors have normal behaviour, however, with the passage of time some nodes show malicious behaviour. We configured some malicious nodes in NS3 by setting the drop probability as different values. The change in drop values allows the packet to be forwarded selectively and making the node malicious. We also simulated a Sybil attack by cloning the same node identity and deploying it in the network simulation. Similarly, we also simulated node insertion

with fake identities. Sensor nodes are responsible for generating User Datagram Protocol (UDP) traffic with a packet size of 50 bytes over a simulation time of 100s and we considered Adhoc On-Demand Distance Vector (AODV) routing protocol for data transmission. During the traffic flow, the trustworthiness and residual energy data of nodes are monitored to detect malicious activity. Furthermore, this data is stored in the blockchain in the form of transactions. The simulator parameters can be found in Table 5.1.

Table 5.1: Parameter settings used in the evaluation.

Parameters	**Setting**
Network Area	100m X 100m
Node deployment	Grid
Number of cluster heads	50
Malicious cluster heads	5,10,15,20
PHY standard	IEEE 802.11
Routing protocol	AODV
Traffic type	UDP
Node energy	100 mJ
Simulation time	100s

5.1.1 Performance Metrics

For our evaluation purposes, we evaluated different performance metrics mentioned below by varying the number of malicious nodes and network size.

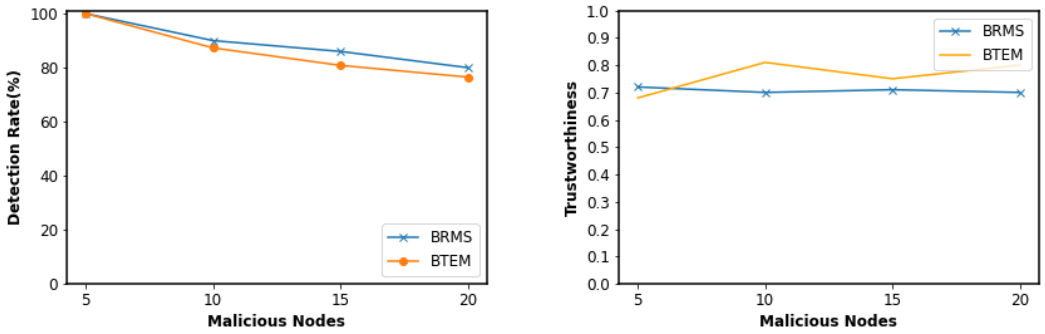
Detection time: It is the time taken by the framework to identify all the malicious nodes present in the network.

Detection rate (DR): The ratio of number of malicious nodes detected (M_d) to the total number of malicious nodes (T_m) is termed as detection rate. This can be computed by the formula given below:

$$DR = \frac{M_d}{T_m} \times 100 \tag{5.1}$$

5.2 Discussion of Results

From Figure 5.1a, we observe that the detection rate of our framework is better than that of BTEM. The detection rate for both the frameworks is dropping as we



(a) (b)

Figure 5.1: Comparison with BTEM model a) Detection Percentage b) Node trustworthiness

increase the number of malicious nodes but still, our framework outperforms BTEM by some margin. The reason behind this performance trend is attributed to our two parameters trust and blockchain-based historical reputation. Contrarily, the BTEM paper only uses one factor to detect the malicious nature of a node, which often results in false positives. Our framework avoids false positives by using historical reputation calculated using blockchain.

In Figure 5.1b, we observe that the trustworthiness of the nodes in BTEM varies with the increase in the number of malicious nodes, whereas our framework shows almost constant trustworthiness for different number of malicious nodes. This variation leads to a lot of false-positive in BTEM as it would be difficult to set a threshold, which can result in classifying normal nodes as malicious and vice-versa. As a result of constant trustworthiness within our framework, we are able to set a precise threshold limit and thus, this could be one of the reasons for having zero false positives in our case.

Figure 5.2 shows malicious node detection over different dropping rates. We see that the performance for 30% and 50% drop rate has a similar number of detection rates. The difference is with respect to the time taken to detect these malicious nodes, it is observed that as the packet dropping rate increases, the framework requires more time to detect the malicious nodes. The underlying cause of this trend is the inability of packets to reach the sink node as the packet drop rate increases. Due to the loss of packets, there are few historical records that permit us to detect whether a node

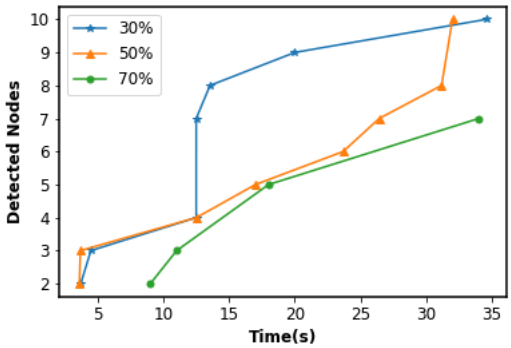
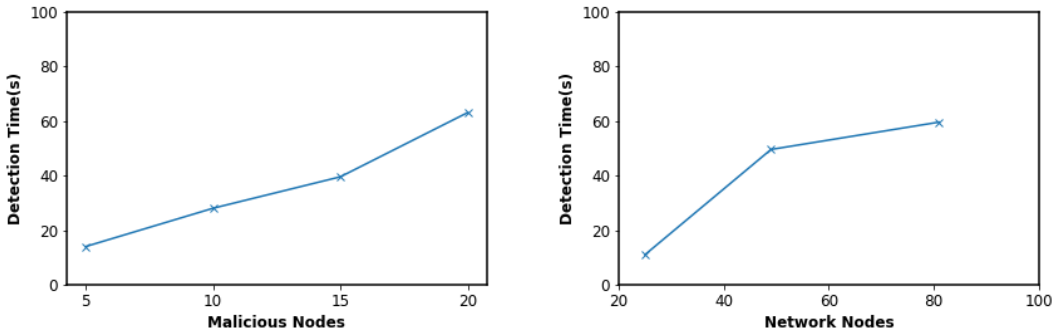


Figure 5.2: Node detection variation over different dropping rate



(a) (b)

Figure 5.3: Detection Time over a) Malicious Nodes Variation b) Network Size Variation

is malicious or not. For the drop rate of 70%, the detection starts slightly late due to the same reason mentioned above and it detects fewer malicious node compared to 30% and 50% drop rates.

In this experiment, we also analysed the detection time when different number of malicious nodes are present in the network. Figure 5.3a shows the increasing trend for detection time as the number of malicious node increases. The nature of this trend is due to the fact that as the malicious node increases, the packet loss in the network also increases, which leads to a fewer number of packets reaching the sink node and hence require more time for gathering the historical records and processing them.

Figure 5.3b also shows a similar trend as the previous graph showing detection time with an increase in network size. The reason is due to an increase in intermediate nodes in the routing path that leads to additional delay for the packet to reach the

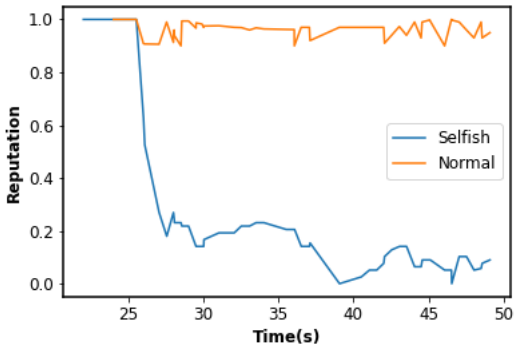


Figure 5.4: Reputation over time for normal and malicious nodes

sink node and hence there is an increase in detection time.

Now we will discuss the performance related to blockchain-based reputation.

Figure 5.4 visualises the behaviour of reputation for both normal and malicious nodes. We can observe that for a normal node, the reputation is constant with fewer fluctuations. On the other hand, the reputation for malicious node decreases rapidly after performing packet drop attacks. This decreasing trend allows us to detect the malicious nodes where reputation is regarded as one of the prime factors.

In Figure 5.5, we show reputation over time for different dropping rates. Here we consider the dropping rates as 30, 50 and 70%. The graph shows reputation decreasing rapidly within a small period for lower dropping rates(30%). Subsequently, the reputation for 50% and 70% drop rates decreased gradually over time. Moreover, we observe that as the drop rate increases, the reputation also decreases gradually over the total simulation time. This is due to the reason that we have fewer prior records as the drop rate increases leading to more processing time, hence we see a slow decrease in reputation factor.

5.3 Security Analysis

In our framework, the majority of the attacks are detected during or after the decryption phase at the sink node. Based on our simulated network, we have performed an analysis for our framework. In this analysis, we demonstrate our framework's usefulness in detecting various kind of malicious attacks. It has also been observed that

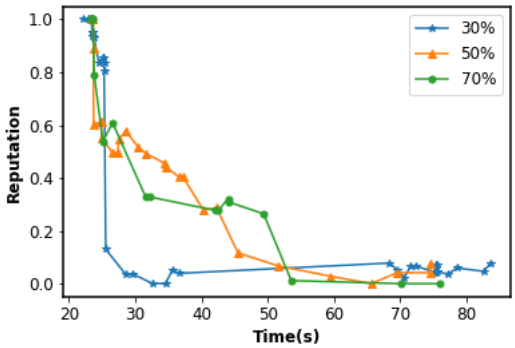


Figure 5.5: Reputation over time at different dropping rates

some of the malicious nodes would remain undetected if they are simultaneously performing another attack with selective forwarding. The reason behind this is when a malicious node performs another attack, it consumes some energy and its reputation does not degrade, leading to a similar reputation as of a normal node. In this case, our framework will consider the corresponding malicious node as a normal node. Now we will explain the attacks with proper reasoning for their detection.

- **Selective forwarding attack** This packet drop attack involves adversary to compromise nodes which shows selfish behaviour by dropping a packet. For every packet drop, the nodes trust would decrease and eventually only consume a small amount of energy in receiving the packet. Our model is designed in such a way that if there is a packet drop, energy consumption will decrease, which will lead to a decrease in reputation over a period. When the reputation is below the predefined threshold, we can say that the node is malicious.

- **Sybil attack** This kind of impersonation attack is detected by the use of cryptographic key pairs provided to each node before node deployment. This attack is exposed when we observe a failure in decrypting the inner part containing energy and trust level of the decrypted tag by the node key. On the other hand, if the same malicious node refuses to forward the packet, our framework will still be able to detect the node because of the reason mentioned in the selective forwarding attack.

- **Replay attack** These attacks are mostly conducted to create traffic congestion in the network by flooding it with the same packets. In order to mitigate this,

we use sequence number or timestamp. The sequence number associated with a data packet ensures its freshness. This provides protection from invalid information getting processed by the sink node. Furthermore, if the sink node observes any repetitive sequence numbers, it generates an alert for replay attacks.

- **Node insertion attack** The framework requires all sensor nodes to get registered first before deployment. These nodes are allocated with the keys and enrollment certificates stored on the blockchain. If a node's identity is not matched with any node certificate present in the registration list, then the sink node detects malicious node insertion in the network.

- **Packet Modification attack** This occurs when a malicious node tries to change the fields of the data packet. In our framework, we allocate keys for encrypting data packets, including the trust and energy fields. Therefore, when communication takes place between nodes, the packets are always in an encrypted format, so any manipulation in the encrypted packets will lead to failure in the decryption process. The sink node will recognise this attack by decrypting the packet using its key. If there is any modification, the sink node won't be able to decrypt the packet tags.

The framework achieves some of the security requirements which are necessary for any application. Our analysis indicates that mainly blockchain technology is used to attain security goals. In addition, our framework uses asymmetric key pairs to accomplish some of the other security goals like:

- **Data Integrity:** Through the blockchain, we have succeeded in maintaining the integrity of all node information through transactions. The nodes record is maintained in an immutable ledger which prevents them from any manipulation. Furthermore, these historical records extracted from the blockchain by the BS are coming from a secured source which allows correct estimation of the reputation for the particular node.

- **Data Availability:** It is one of the main characteristics of blockchain since it offers distributed storage of data. Each block node has a copy of data, so even if one blockchain node gets compromised or fails, our data is secured. Thus, this allows the stability and reliability of the system.

- **Data Confidentiality:** We maintain the confidentiality of data in two ways. Firstly, we use encryption for data packets which provides security against attacks such as data interception. Secondly, we rely on a permissioned blockchain that allows only the base stations to read the node information. This ensures confidentiality as data is accessed only by designated stations.

- **Data Authenticity:** Currently, the framework works only with pre-authenticated validator nodes. These nodes are predefined to perform data validation and verification, thereby ensuring that blocks created and the data stored on the blockchain are from authentic sources.

5.4 Blockchain Model Analysis

Blockchain network is also vulnerable to various attacks such as 51% attack and DoS attacks. The 51 percent attack is an attack on a blockchain consensus mechanism where the adversary gains control of 51 percent of the validator nodes or computational power to destroy the transaction validation process. In the case of proof of work consensus, the overall reliability of the blockchain can be affected since 51% of computational power can override the consensus algorithm results. Similarly, PoA consensus algorithm results can be overturned when attackers take control over 51 percent of validator nodes. In our framework, we have utilised the PoA consensus mechanism for permissioned blockchain where all the validators are pre-authenticated and adding a new validator node is difficult as we have to preconfigure it before adding it to the blockchain network. Furthermore, the PoA consensus is independent of the computational power, even if the attacker takes control of computational power, they cannot carry out the 51 percent attack. To launch this attack on the PoA blockchain, it is necessary to gain control of 51 percent of validating nodes which is extremely difficult on the permissioned blockchain.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Wireless sensor networks face many security threats during data transmission leading to the disruption of data availability. To counter this problem, we have proposed a decentralized framework for reputation management and malicious node detection. The nodes which hamper the data routing through selective forwarding of data packets are discovered by estimating their reputation through blockchain and trust from child nodes. The designed framework begins with the clustering and data transmission phase followed by the malicious node detection phase. The implemented scheme provides several advantages like node identity and key management in a decentralized manner, the authenticity of intermediate nodes, blockchain-based historical reputation approximation and identifying various kind of attacks. Additionally, it also accomplishes the majority of the security goals like data integrity, data authenticity and data availability.

In this work, we assign a reputation to each node as the data routing progresses. The reputation for malicious nodes allows us to determine the malicious activity carried by a node. The reputation of a node is calculated using a nonlinear function which over a while falls rapidly for node showing selfish behaviour. Besides, using a PoA consensus mechanism between validators rather than traditional consensus like PoW proved to be beneficial for low-powered sensor devices where transactions are validated by fewer validators before committing to the blockchain. The proposed model has been simulated on NS3 and compared with an existing model to evaluate its performance metrics like detection rate, detection accuracy and detection time. The security analysis represents an enhancement in recognizing the malicious behaviour and the reason behind it is using two parameters that prevent any false positives. The results reveal the effectiveness of our proposed framework. We propose to use this framework as a security measure for applications covering large areas such as smart

farming, smart cities etc.

6.2 Future work

The proposed framework has a few limitations. Firstly, we did not consider a scenario where a node failure happens. Secondly, because we are using the blockchain for reputation management, the size of the blockchain will increase as time progresses, which can create memory constraint at the base station. This work can be extended to overcome these limitations by keeping track of the state of nodes and maintaining a separate registry for it. So in the event of a node failure, the base stations can temporarily remove node ID from its lookup list. To address the blockchain memory issue, we can implement some efficient memory management techniques to free up some of its space. For the current scope of research, we carried out simulations using Wifi based radio, however, in future, we would like to analyse our framework for low power protocols like the Routing Protocol for Low Power and Lossy Networks (RPL) and evaluate its efficiency. Furthermore, we have considered static nodes for the current implementation, it would be an interesting research work to implement the proposed framework on mobile devices and investigate its performance.

Bibliography

[1] Proof of stake. https://www.investopedia.com/terms/p/proof-stake-pos.asp, 2019. (Last accessed 04-April-2021).

[2] Proof-of-authority consensus. https://apla.readthedocs.io/en/latest/concepts/consensus.html, 2020. (Last accessed 04-April-2021).

[3] M. I. Channa A. Ahmed, K. A. Bakar and A. W. Khan. A survey on trust based detection and isolation of malicious nodes in ad-hoc and sensor networks. *Frontiers of Computer Science*, 9(2):280–296, 2015.

[4] E. Basan A. Basan and O. Makarevich. Development of the hierarchal trust management system for mobile cluster-based wireless sensor network. In *In Proceedings of the 9th International Conference on Security of Information and Networks*, SIN '16, page 116–122, 2016.

[5] M. Abidalrahman. Energy efficient security for wireless sensor networks. 2013.

[6] S. Asiri and A. Miri. A sybil resistant iot trust model using blockchains. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Halifax, NS, Canada, June 2019.

[7] A. Beheshtiasl and A. Ghaffari. Secure and trust-aware routing scheme in wireless sensor network. *Wireless Pers. Commun.*, 107(4):1799–1814, Aug.

[8] M. Z. A. Bhuiyan and J. Wu. Collusion attack detection in networked systems, 2016.

[9] M. A. A. Careem and A. Dutta. Sensechain: Blockchain based reputation system for distributed spectrum enforcement. In *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019.

[10] NS3 community. Ns3 network simulator. https://www.nsnam.org/, 2011. (Last accessed 04-April-2021).

[11] L. Conway. Blockchain explained. https://www.investopedia.com/terms/b/blockchain.asp, 2020. (Last accessed 04-April-2021).

[12] A. Forster. *Introduction to Wireless Sensor Networks*. IEEE, 2016.

[13] M. Großmann H. L. Cech and U. R. Krieger. A fog computing architecture to share sensor data by means of blockchain functionality. In *IEEE International Conference on Fog Computing (ICFC)*, pages 31–40, 2019.

[14] A. Kar H.Sarma and R. Mall. A hierarchical and role based secure routing protocol for mobile wireless sensor networks. *Wireless Personal Communications*, 90(3):1067–1103, 2016.

[15] V. P. Illiano and E. C. Lupu. Detecting malicious data injections in wireless sensor networks. *ACM Comput. Surveys*, 48(2):1–33, 2015.

[16] G. Ibbotson J. Marchang and P. Wheway. Will blockchain technology become a reality in sensor networks? In *2019 Wireless Days (WD)*, pages 1–4, 2019.

[17] Y. Xu L. Chen J. Yang, S. He and J. Ren. A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks. *Sensors*, 19(4), Feb 2019.

[18] T. Kim et al. A novel trust evaluation process for secure localization using a decentralized blockchain in wireless sensor networks. *IEEE Access*, 7:184133–184144, 2019.

[19] E. Irmak M. M. Ozcelik and S. Ozdemir. A hybrid trust based intrusion detection system for wireless sensor networks. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, Oct 2017.

[20] M. Chatterjee M. Salimitari and Y. P. Fallah. A survey on consensus methods in blockchain for resource-constrained iot networks. *Internet of Things*, (4):1–19, 2020.

[21] D. Milkovich. 15 alarming cyber security facts and stats. `https://www.cybintsolutions.com/cyber-security-facts-stats/`, 2020. (Last accessed 04-April-2021).

[22] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[23] V. Jain P. Dewal, G.S. Narula and A Baliyan. Security attacks in wireless sensor networks: A survey. *Cyber Security Advances in Intelligent Systems and Computing*, 729:47–58, 2018.

[24] K. Parmar and D. C. Jinwala. Concealed data aggregation in wireless sensor networks: A comprehensive survey. *Comput. Netw.*, 103:207–227, Jul 2016.

[25] M. Abdelrazzak R. E. Mohamed, A. I. Saleh and A. S. Samra. Survey on wireless sensor network applications and energy efficient routing protocols. *Wireless Pers. Commun.*, 101(2):1019–1055, 2018.

[26] F. Outay A. Yasar R. W. Anwar, A. Zainal and S. Iqbal. BTEM: Belief based trust evaluation mechanism for wireless sensor networks. *Future Gener. Comput. Syst.*, 96:605–616, Jul 2019.

[27] A. Saidi and K. Benahmed Pr. Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks. *Ad Hoc Networks*, 106, 2020.

[28] N. Singh. Permissioned vs permissionless blockchains. `https://101blockchains.com/permissioned-vs-permissionless-blockchains/`, 2020. (Last accessed 04-April-2021).

[29] B. Sun and D. Li. A comprehensive trust-aware routing protocol with multi-attributes for wsns. *IEEE Access*, 6:4725–4741, 2018.

[30] Y. Sun and Y. Zhao. Dynamic adaptive trust management system in wireless sensor networks. In *IEEE 5th International Conference on Computer and Communications (ICCC)*, Apr 2020.

[31] H. Farhat T. Azzabi and N. Sahli. A survey on wireless sensor networks security issues and military specificities. In *017 International Conference on Advanced Systems and Electric Technologies (IC$_A$SET), pages*$66 - -72, 2017$.

[32] P. D. Shenoy U. Prathap and K. R. Venugopal. CMNTS: Catching malicious nodes with trust support in wireless sensor networks. In *2016 IEEE Region 10 Symposium (TENSYMP)*, ICC '05, July.

[33] G. D. Putra A. Dorri V. Dedeoglu, R. Jurdak and S. S. Kanhere. A trust architecture for blockchain in iot. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '19, Nov 2019.

[34] H. Wu W. Alghamdi, M. Rezvani and S. S. Kanhere. Routing-aware and malicious node detection in a concealed data aggregation for wsns. *ACM Trans. Sensor Netw.*, 15:18.

[35] Z. Tian J. S. Chen B. Wang W. She, Q. Liu and W. Liu. Blockchain trust model for malicious node detection in wireless sensor networks. *IEEE Access*, 7:38947–38956, March 2019.

[36] E. Wheller. *Security risk management: Building an information security risk management program from the Ground Up*. Elsevier, 2011.

[37] L. Die Yang, Guang and Z. Wei. Challenges, threats, security issues and new trends of underwater wireless sensor networks. *Sensors*, 18(11):3907, 2018.