ISSN: 2182-2069 / E-ISSN: 2182-2077

Malicious Nodes Detection and Avoidance Using Trust-based Routing in Critical Data Handling Wireless Sensor Network Applications

B. Sreevidya[1*], and Dr.M. Supriya[2]

[1*]Department of Computer Science & Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. b_sreevidya@blr.amrita.edu, https://orcid.org/0000-0002-0876-3307

[2]Department of Computer Science & Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. m_supriya@blr.amrita.edu, https://orcid.org/0000-0001-6147-7142

Received: March 16, 2024; Revised: May 17, 2024; Accepted: June 20, 2024; Published: August 30, 2024

Abstract

With the ever-evolving wireless technological domain, large number of applications are developed which utilizes the advantages of wireless technology. Healthcare, defense, intrusion detection, agriculture, logistics etc. are examples of domains which witnessed a complete paradigm shift due to the incorporation of wireless technology. Wireless Sensor Networks (WSN) have provided the most suitable platform for developing applications in various domains utilizing the advantageous of wireless transmission of data between devices. Along with the numerous advantages offered by the wireless communication technology, applications developed over WSN faces the most critical challenge of managing the battery for performing the various operations including computations and communication. Due to this constraint, generally applications deployed over WSNs are focused on optimizing energy consumption to improve the lifetime of the WSN nodes. But when WSNs are used to deploy applications from defense or healthcare sectors, an additional requirement has been taken the focus which is data security. Most of the applications in healthcare or defense sectors handle confidential data and communication of such data must be secured. The traditional security schemes deployed for secured data communication are computationally complex and consumes larger amount of energy this making it less feasible for WSN based applications. This opens the possibility of attacks on WSN based applications. Cardinal impairment is regarded as attacks emanating from compromised nodes and changing data. The paper analyses several attacks and proposes a new security approach to prevent data change due to compromised nodes. The suggested technique is a non-cryptographic scheme which is computationally less complex. In this scheme, nodes compute trust of the neighbors and exclude hostile nodes from the probable list of forwarding nodes based on low trust value. Trust computation is carried out considering various parameters like packet drop rate, packet rejection ratio and remaining energy level of the node. The proposed system is simulated using Network Simulators (NS2) and performance of the proposed system is compared against traditional schemes.

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 3 (August), pp. 226-244.
DOI: *10.58346/JISIS.2024.I3.013*

*Corresponding author: Department of Computer Science & Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India.

Keywords: Wireless Sensor Networks, Trust-based Routing, Data Integrity Attacks, Network Lifetime.

1 Introduction

Sensors and wireless communication have opened up enormous possibilities for designing more competent and intelligent apps to optimize resource efficiency and give end-users better features. Applications from diverse areas, such as surveillance, health care, entertainment, are a few examples of how wireless technology has been leveraged for efficient end-user services. Wireless Sensor Networks (WSN) are sensor nodes that communicate wirelessly. The WSN's wireless medium enables sensor nodes to be mobile and the WSN to be scalable. Wireless sensor networks' inherent mobility and scalability encouraged developers to create deployable applications in environments with little human interaction.

Replacing or assisting with applications that require human mediation has opened up a broad avenue for research into the safety of the human species operating in adverse settings. WSNs have replaced human beings in various applications, including structural health monitoring and mines exploration. WSNs have transformed the perspective of application developers in border monitoring. Another arena where remote patient monitoring, assistive technologies for the elderly, and programs that do primary level medical diagnostics are developed is health care.

Researchers have optimized energy consumption, deployment tactics, and faster processing in WSNs in order to create more innovative and cost-effective solutions. As the nodes are deployed in unsupervised environments, energy is regarded as a critical resource in the sustenance of the WSNs. The energy source connected to the sensor nodes is used wisely to extend the network's lifetime and application. WSN energy consumption can be optimized in a variety of methods. One approach is to keep the sensor nodes in idle/sleep mode when they are not in use; one approach is to keep them in idle / sleep mode.

Unlike its active form, which consumes energy. A variant of the previously discussed system employs energy-optimized routing algorithms (Naveenraj & Student, 2017). Creating energy-efficient algorithms for data processing is another way to reduce energy use.

As previously said, several sectors, such as surveillance, patient monitoring, intrusion detection, handle massive amounts of data, and processing this data must be done in real-time. As a result, secure data transmission across nodes in the WSN is also critical. WSN attacks come in many forms, and most of them are based on the concept of compromised nodes or malicious modes. A malicious/compromised node that controls an attacker can harm the system. Among the several malicious node assaults, the False Data Injection attack is the most dangerous. A hacked node either injects bogus data into the network or alters the data of ongoing communication in False Data Injection attacks. As a result, a rogue node can disrupt the routing process, rendering data communication not secure. A rogue node can also shorten the lifespan of a WSN by flooding the network with traffic.

Given that most applications are data essential addressing data security in transmission and improving energy usage (Sreevidya & Rajesh, 2017) among WSN nodes is critical. Cryptographic techniques are the most frequent strategy for introducing security, although the most efficient cryptographic algorithms are computationally complex and need much energy (Al Shehri, 2017). As a result, if the goal is to reduce energy usage, cryptographic solutions for data security are out of the

question. As a result, developers have frequently overlooked the necessity for data security to accomplish energy optimization (Sreevidya & Rajesh 2018).

Non-cryptographic techniques are viable WSNs for achieving data security and optimizing energy utilization (Sreevidya & Rajesh, 2018). The use of the trust factor to eliminate compromised nodes from the WSN is a non- cryptographic strategy that is resistant to most attacks and adds to the efficient use of energy in the nodes. Furthermore, the trust factor can be employed at several levels, such as a decision-making parameter in data forwarding to determine which nodes need to be considered for data aggregation further to determine the path along which the gateway can be accessed.

Data security is achieved through traditional cryptographic schemes like authentication and encryption techniques, but when a node itself compromises to malicious activities, such schemes are ineffective. To counter the attacks initiated by malicious nodes or to identify malicious nodes in WSN, trust factor is identified to be the most effective solution and the fact that techniques using trust factors are computationally lesser complex compared to traditional cryptographic schemes makes it more advantageous in the light of energy optimization constraints in WSNs. Trust factor-based schemes can counterattack like black holes, wormholes, sinkholes, grey holes, Sybil attacks, and on–off attacks. Nodes compute trust by itself as well as by its neighbors. Identifying the trust score of a node will help in judging whether the node is trustworthy or malicious. If the node is categorized as malicious, neighboring nodes will avoid the malicious nodes while forwarding the data packets. This way, the impact of attacks generated by malicious nodes can be minimized and reliability and throughput can be improved.

The trust factor of a node is computed in two different ways in WSNs. Centralized approach in which the Base Station (BS) computes the trust value of each node in the network while in distributed approach, each node computes its trust by itself considering various parameters and the trust surrounding nodes exhibit. Single point failure is the drawback of centralized approach and thus, most of the WSNs apply distributed approach for trust computation. The proposed work focuses on establishing a security mechanism that computes the trust factor of all surrounding nodes and uses it in the routing process (Kasar et al., 2015). Because the proposed method of trust computation requires considering additional parameters, it outperforms other similar approaches. Furthermore, the suggested technique is unique in that weightage for each parameter is considered for the trust value computation.

The remainder of the paper summarizes earlier work in the same domain, followed by a complete description of the proposed system with appropriate diagrammatic representations. Following the proposed system description, simulation details and results are addressed, and the conclusion examines the results gained and future research avenues feasible in the offered solution.

2 Related Works

Wireless Sensor Networks are widely utilized in various applications, including surveillance and healthcare (Bade & Garba, 2019). The security of data transmitted through the WSN is critical in such domains. Maintaining data confidentiality and integrity are two critical elements to consider in WSN (Pawar & Agarwal, 2017). Extensive research is being conducted to provide mechanisms for secure data transmission via WSNs. Understanding the numerous security threats to WSNs is required to develop a safe data transfer in WSN (Chelli, 2015).

Wormhole Assaults, Sleep Deprivation Attacks, False Data Injection Attacks, and other security threats or attacks discovered in the literature survey are Wormhole Attacks, Sleep Deprivation Attacks,

and so on. A detailed analysis of several application domains in which WSN can be exploited is also carried out to understand the impairment of these assaults. Furthermore, threats are classified depending on the network levels they affect to comprehend the risks and the solutions offered. As a result, the study concluded that the data connection and network layers are particularly vulnerable to the bulk of WSN assaults.

Two often seen attacks - Sleep Deprivation Attack and Wormhole Attack – were researched and the defenses. One of the proposals suggests building a protocol to mitigate the effects of a sleep deprivation assault with minimal protocol overhead. Similarly, to avoid wormhole assaults, the AODV protocol is improved by including a node's trust as a decision parameter when forwarding packets.

The literature research revealed that compromised nodes are responsible for the bulk of high magnitude assaults in WSN (Choi & Song, 2006). The idea of the node trust factor (Dhivvya et al., 2017) is widely acknowledged in most research to identify and eliminate compromised nodes from decision-making. Based on this viewpoint, a survey of the literature on articles that work on trust factor computation (Simi & Ramesh, 2019) was undertaken. One study proposes a novel routing strategy that uses the trust factor and residual battery information for routing. Another research (Oke et al., 2018) introduces two-level trust computation to make an Intrusion Detection System (IDS) safer. One level of analysis is performed at the cluster head, while another level is performed at the base station.

According to the literature review, the most common attacks in WSN have compromised/malicious nodes. As a result, it is critical to devise a mechanism that prevents compromised/malicious nodes from impacting the system. According to the literature review, among the existing techniques for dealing with compromised/malicious nodes, cryptographic schemes consume more energy and are not recommended for WSN (Dharini et al., 2020). Using trust value improves energy usage and reliability against compromised/malicious nodes in many non-cryptographic ways. It is also observed that prior trust calculation systems examine only a few parameters to determine trust, whereas many more network parameters influence a node's trust value. To cope with threats created by compromised/malicious nodes, implementing a system that leverages trust value for eliminating compromised/malicious nodes using additional network characteristics (both direct and indirect) will be advantageous.

3 Proposed Model

The proposal is to create a safe and energy-efficient data transfer mechanism. According to the literature review, cryptographic approaches for data security are not recommended because they are computationally complex and thus inefficient. As a result, the proposed approach is a non- cryptographic scheme that eliminates assaults created by compromised nodes by leveraging the trust value of WSN nodes. The suggested technique is modularized into Node Deployment, Trust Value Computation, Trusted Neighbors List, Trust-based Routing, and Secured Data Transfer. It is always easy to design and upgrade a modular system.

The proposed system's block diagram is shown in Figure 1. Nodes are initially placed; the nodes then identify their neighbors and build the network. Next, nodes calculate their trust value and communicate it with their neighbors. Finally, each node receives the trust values of its neighbors and compiles a list of trusted neighbors.

The routing module will pick the best path to the destination and the next hop to which the packet must be delivered once the trustworthy neighbor's list has been generated. After successfully identifying

the next hop, a secured data transfer phase will begin, in which data transfer between nodes is implemented using a simple cryptographic technique.

The control flow of the suggested system is depicted using a flowchart in Figure 2.

The proposed system is computationally simple and uses less energy while operating. As a result, the network's lifetime is extended.

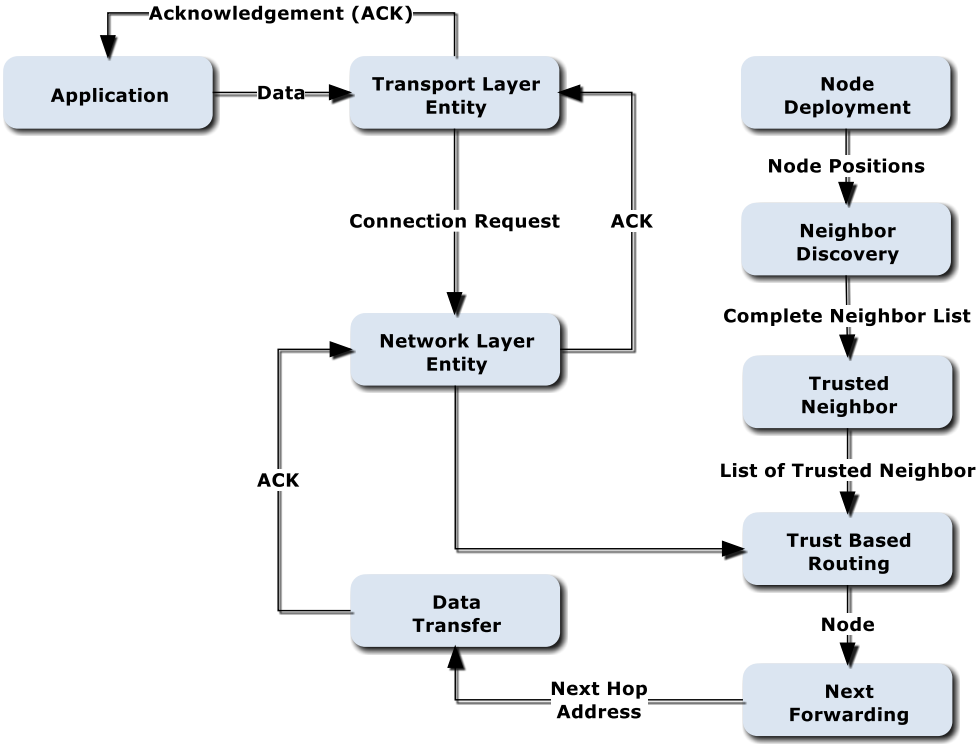

Figure 1: Block Diagram

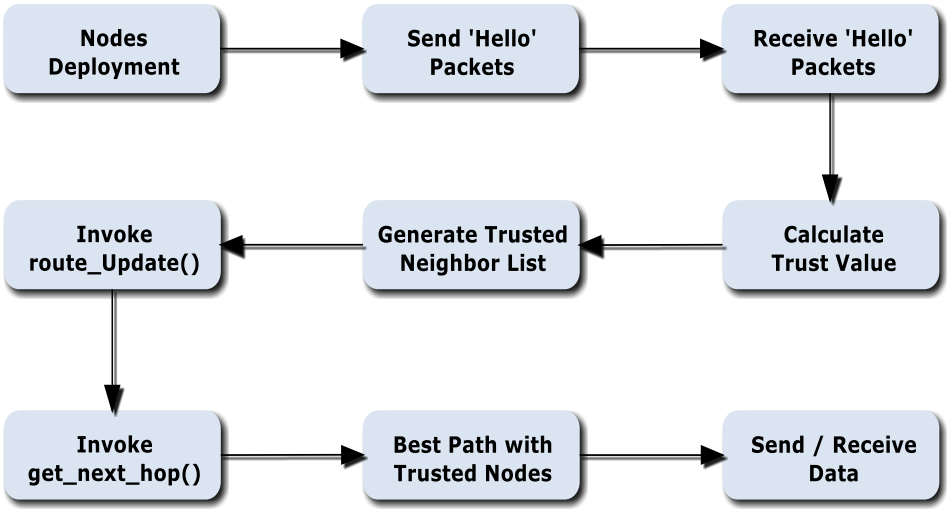

Figure 2: Flow Chart

4 Implementation

A Simulation tests the proposed system and compares its performance to existing systems. Network Simulator Version 2 (NS2) is used to simulate the proposed method in a WSN environment. NS2 includes deploying a wireless sensor network and conducting experiments with various protocols at different network layers. The following submodules are included in the proposed system (Zhong et al., 2016).

- Node Deployment
- Trust Value Computation
- Trusted Neighbor Listing
- Trust based Routing
- Secured Data Transfer

4.1. Node Deployment

NS2's random deployment feature is used to deploy nodes at random. A different number of nodes is deployed to understand the performance of the proposed scheme. The simulation experiment is carried out with 20, 40, 60, 80, and 100 nodes in the same study area.

4.2. Trust Value Computation

The trust value is a critical parameter in selecting a node to participate in the routing process. Several parameters are taken into account to calculate a node's trust value. The parameters taken into account for the computation are the remaining battery/energy, the number of packets dropped, the number of packets forwarded, the delay, the sleep cycle time, and the reputation (Neighbor provides this information). These parameters are weighted in the computation of the trust value.

4.3. Trusted Neighbor Listing

In a normal routing process, the trust value computed at each node is shared and embedded in echo packets, primarily used to identify neighboring nodes. This operation assists each node in discovering its neighboring nodes as well as their corresponding trust values. Eq (1) generates the Neighbor list by selecting neighbor nodes whose trust value is greater than the threshold value. The threshold value is then calculated experimentally.

$$Neighbor(N_x, N_y) = \begin{cases} 1 & \left((T_x \; and \; T_y) > \; T_{thr}\right) \\ 0 & \left(distance \; between \; N_x \; and \; N_y\right) < \; D_{thr} \\ & else \; both \; are \; NOT \; Neighbors \end{cases} \quad (1)$$

Where,

$N_x \; and \; N_y$ are two random nodes

$T_x \; and \; T_y$ Trust values of $N_x \; and \; N_y$ nodes

T_{thr} – Threshold of Trust Value

D_{thr} – *Distance Threshold for Neighbor Node*

4.4. Trust based Routing

The routing table is created based on the trusted neighbor list created in the previous step. This routing table is shared with the neighbors regularly. This routing scheme differs from the most common AODV routing, which uses echo packets to identify neighboring nodes and the hop count. However, along with the information above, the proposed routing scheme. This shared trust value is used as a parameter in the node's trust value computation.

4.5. Secured Data Transfer

When a sensor node needs to communicate sensitive data to another sensor node or base station, the request is sent to the transport layer, forwarding it to the network layer to complete the routing process and determine the best path (Awais & Gouri, 2017). When the network layer receives the request, it must compute the trust value, and trust-based routing is performed based on the trust values of neighbors. The trust- based routing process produces the best path, and the node can forward the packet to the next trusted neighbor.

4.6. Trust Value Computation of a Node

The term "trust" refers to the degree of reliability in a sensor network. The common understanding of one node's trust in another specific node is not jeopardized. Researchers have used a variety of indices to measure/represent a node's trust value. In some approaches, the computation includes examining previous transactions performed by the node. The proposed scheme computes the trust value based on the remaining battery/energy, the number of packets dropped, the number of packets forwarded, the delay, the sleep cycle time, and the reputation (Neighbor provides this information). Trust value is computed regularly because it is time-dependent and can fluctuate based on the node's transactions.

The parameters used to compute a node's trust value are determined by how the trust value is defined. In the proposed scheme, trust is regarded as a measure that describes the likelihood that a node is not a malicious node. Increase the node's trust value. Reduce the likelihood that the node is malicious. This probability is measured in suspicious actions taken by a node and identified using the node's energy consumption. Assume, for example, that the node consumes much energy compared to other nodes. This may be due to suspicious activity, which identified the node as a compromised/malicious node. The following parameters are determined to influence a node's trust value based on an analysis of energy consumption. Each parameter is assigned a weight that will be used in calculating the trust value.

- Remaining Battery Energy (E_r)
- No. of packs Forwarded (N_r)
- Sleep Cycle time (T_s)
- Reputation (R)
- No. of packets Dropped (N_d)
- Delay (T_d)

The first four parameters positively influence trust, while the last two negatively influence trust.

Trust value T of a node is computed as follows Eq (2, 3):

$$T = T_{prev} + \big((W_1 * E_r) + (W_2 * N_r) + (W_3 * T_s) + (W_4 * R)\big) - \big((W_5 * N_d) + (W_6 * T_d)\big) \quad (2)$$

Where,

$$W_1 + W_2 + W_3 + W_4 + W_5 + W_6 = 1 \; and \; (W_1, W_2, W_3, W_4, W_5, W_6) > 0 \quad (3)$$

These weight values are determined by the energy consumption rate of these parameters and their impact on the node's lifetime. The node's energy is analyzed, and the percentage of influence is identified as the weight.

Along with this, we also incorporate an integrated trust management model which also considers direct and indirect trust calculation. Below given figure 3 delineate a basic WSN model for direct and indirect trust calculation.

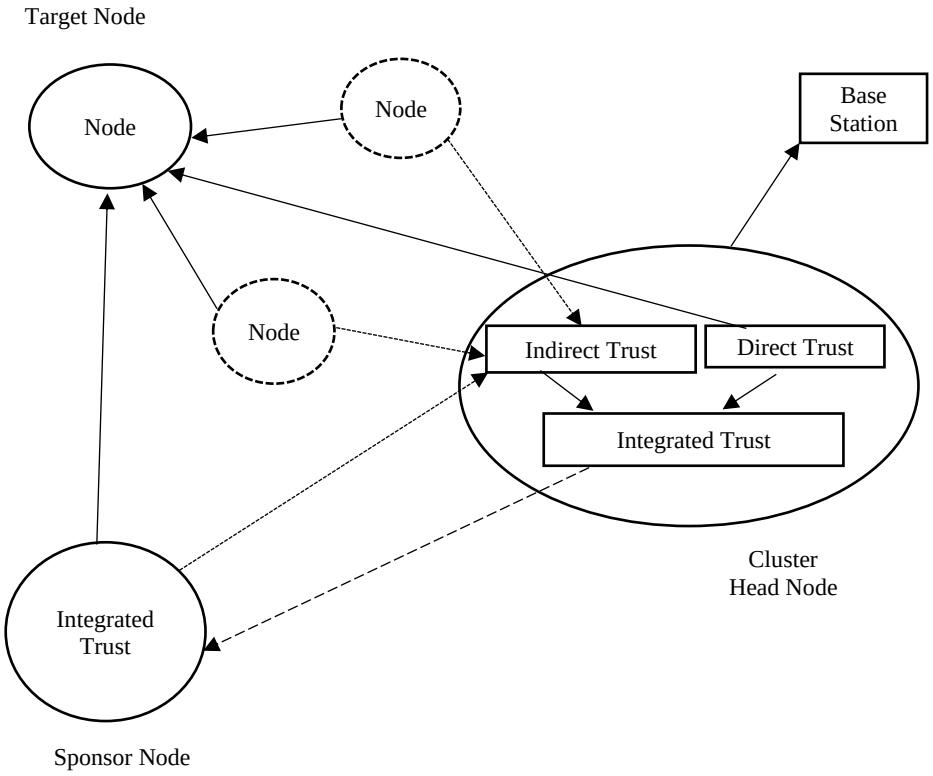

Figure 3: WSN Direct and Indirect Trust Model

Here, we assume that the cluster head has higher computational resources compared to other nodes. Also, the base station is considered to be completely trustworthy. Here, we consider node level and cluster level trust management schemes. The node level trust management also includes direct and indirect trust management process.

4.7. Node-level Trust Management

The direct trust value between node s and t is denoted as $T_{s,t}$. The trust value for a cluster is denoted as G_t and an integrated trust value is given as I_t which is calculated based on direct trust value of cluster node and group trust value. In order to compute the direct trust value, we use behavioral trust evaluation, we consider node closeness $D_i^{closeness}$, honesty of the node $D_i^{honesty}$, intimacy $D_i^{intimacy}$, and interaction frequency of the node $D_i^{interaction}$.

- **Closeness $D_i^{closeness}$:** it is a trust matrix which is used to evaluate the total number of sensor nodes which are covered in beacon node in one-hop neighbor. It is computed as Eq (4):

$$D_i^{closeness} = \frac{\sum_{j=1,i\neq j}^{N} D_{ij}^{one-hop}}{\sum_{j=1,i\neq j}^{N} D_{ij}^{max-hop}} \quad (4)$$

$\sum_{j=1,i\neq j}^{N} D_{ij}^{one-hop}$ denotes the one-hop neighbor node which is covered by the i^{th} beacon node and $\sum_{j=1,i\neq j}^{N} D_{ij}^{max-hop}$ denotes the all sensor nodes in the network, excluding i^{th} beacon node.

- **Honesty $D_i^{honesty}$:** it is the measurement of successful and unsuccessful interactions among sensor nodes and the value of honesty varies within [0,1]. This can be calculated as Eq (5):

$$D_i^{honesty} = \frac{I_{i,j}^{successful}}{I_{i,j}^{successful} + I_{i,j}^{unsuccessful}} \quad (5)$$

The $I_{i,j}^{successful}$ and $I_{i,j}^{unsuccessful}$ denotes the successful and unsuccessful interaction between node i and j.

- **Intimacy $D_i^{intimacy}$:** it is a matrix to measure the duration of interaction of nodes. The value of $D_i^{intimacy}$ represents the interactions of durations of i^{th} node. It can be represented as Eq (6):

$$D_i^{intimacy} = \frac{t_{ij}}{t_{ij} + t_{ik}} \quad (6)$$

Here, t_{ij} and t_{ik} denotes the total duration of interaction of node i with node j and k respectively.

- **Interaction of frequency $D_i^{frequency}$:** it is a matrix to evaluate the total number of interactions between beacon nodes Eq (7).

$$I_i^{frequency} = \frac{n_{ij}}{n_{ik}} \quad (7)$$

n_{ij} is the interaction between beacon node and n_{ik} is the total number of interactions with other nodes.

The overall trust value is expressed as Eq (8):

$$Trust(i) = w_1 \times D_i^{closeness} + w_2 \times D_i^{honesty} + w_3 \times D_i^{intimacy} + w_4 \times I_i^{frequency} \quad (8)$$

Here, $w_1 + w_2 + w_3 + w_4 = 1$. The direct trust value is computed based on these attributes and their cooperation is recorded as presented in table 1. Each attribute has two values which represent the success of attribute and cumulative cooperation.

Table 1: Attributes Contribution for Trust Computing

Attribute	Number of success	Cumulative cooperation
A_1	S_1	CC_1
A_2	S_2	CC_2
...	...	...
A_n	S_n	CC_n

Based on these records, the trust value for attributes can be computed as follows Eq (9):

$$t_{A_i} = \left[100 \times \frac{S_i}{CC_i}\right] \tag{9}$$

The cumulative cooperation values are used for cluster level trust management i.e., higher values of trust show the better stability of clusters. The cluster head trust values are given as Eq (10):

$$T_{ch} = \begin{bmatrix} T_{1,2} & T_{2,1} & \dots & T_{n,1} \\ T_{1,3} & T_{2,3} & \dots & T_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1,n} & T_{2,n} & \dots & T_{n,n-1} \end{bmatrix} \tag{10}$$

Similarly, the group trust matrix is expressed as Eq (11):

$$T_G = \begin{bmatrix} G_1 & T_{ch,1} & \dots & T_{b,1} \\ G_i & T_{ch,i} & \dots & T_{b,i} \\ \vdots & \vdots & \vdots & \vdots \\ G_n & T_{ch,n} & \dots & T_{b,n} \end{bmatrix} \tag{11}$$

The group trust can be computed as follows Eq (12):

$$G_i = \frac{\prod_{j=1,i-1,i+1,..n} T_{j,i}}{\prod_{j=1,i-1,i+1,..n} T_{j,i} + \prod_{j=1,i-1,i+1,..n}(1 - T_{j,i})} \tag{12}$$

Based on these trust values, the integrated trust value can be obtained as Eq (13):

$$I_i = G_i \times W_{group} + T_{ch,i} \times W_{ch} + T_{b,i} \times W_{base} \tag{13}$$

Similarly, we consider indirect trust management model which handles the scenario when extra nodes are deployed in the network. This becomes a scenario of uncertainty to evaluate the trust. Thus, we adopt the Entropy theory where entropy of a random variable μ with a probability mass function $\Phi(\mu)$, is expressed as Eq (14):

$$H(\mu) = -\sum \Phi(\mu) \log_2\big(\Phi(\mu)\big) \tag{14}$$

The entropy of recommendation can be expressed as Eq (15):

$$H(R) = -R \log_2 R - (1 - R) \log_2 (1 - R) \tag{15}$$

At this stage, let us consider a node which has information regarding its own successful and unsuccessful interaction. The entropy values for these observations can be expressed as Eq (16, 17, 18):

$$R^x \alpha_{ij} = \frac{2 \times \alpha_{ij} \times \alpha_{xj}}{(\beta_{ix} + 2) + \big(\alpha_{xj} + \beta_{xj} + 2\big) + \big(2 \times \alpha_{ij}\big)} \tag{16}$$

$$R^x\alpha_{ij} = \frac{2 \times \alpha_{ix} \times \beta_{xj}}{(\beta_{ix} + 2) + (\alpha_{xj} + \beta_{xj} + 2) + (2 \times \alpha_{ix})} \quad (17)$$

$$R_{ij}^x = \frac{(R^x\alpha_{ij} + 1)}{(R^x\alpha_{ij} + R^x\beta_{ij} + 2)} \quad (18)$$

Based on (12), the entropy of R_{ij}^x can be expressed as Eq (19):

$$H(R_{ij}^x) = -R_{ij}^x \log_2 R_{ij}^x - (1 - R_{ij}^x) \log_2 (1 - R_{ij}^x) \quad (19)$$

Based on the entropy, the weights are computed as follows Eq (20):

$$w_x = \frac{1 - \frac{H(R_{ij}^x)}{\log_2 R_{ij}^x}}{\sum_{x=1}^{x=n} \left(1 - \frac{H(R_{ij}^x)}{\log_2 R_{ij}^x}\right)} \quad (20)$$

Finally, the indirect trust values can be given as Eq (21):

$$R_{ij} = \sum_{x=1}^{x=n} (w_x \times R_{ij}^x) \quad (21)$$

Thus, we obtain the direct and indirect trust values for incorporating the security in the network.

5 Results and Analysis

The approach is implemented with NS2 using the Simulation parameters listed in Table 2 to understand the effectiveness of the proposed system. Because there are a few network simulators available to simulate network protocols and related procedures. It is preferable to simulate the proposed system rather than implement it on a real-world WSN. Network Simulator (NS2) is the most popular simulator because it is open source and free to download and use. Furthermore, NS2 includes many libraries that can support the simulation of any network and any network protocols/procedures.

Table 2: The Simulation Parameters

Parameters	Values
Simulation time	25 s
Monitoring area	100 × 100 m2
Number of nodes	20 to 100
Propagation model	Two Ray
Packet interval	0.5 s
Length of the data packet	1000 bytes
Initial energy	100 J
Transmit power	0.9 w
Receive power	0.8 w
Idle power	0.1 w
Sense power	0.0175 w
Routing protocol	AODV (TBR)
MAC layer protocol	IEEE 802.11

The features for deploying wireless sensor nodes, the availability of AODV protocol source code, trace files for monitoring, and log generation made NS2 the best simulator for implementing the proposed system. A random deployment model available in NS2 deploys the sensor nodes. Nodes are initialized with preliminary battery energy and trust value upon deployment. Following this thread, nodes will function as regular sensor nodes, with sleep and active cycle, and data will be collected from the sensors attached during the active cycle. Each node attempts to understand the network to form and communicate the data collected. In traditional routing algorithms, nodes collect data about neighbors by sending Hello packets, and the network is built based on the responses.

The proposed system defines two functions, $send\_hello()$ and $recv\_hello()$, for sending and receiving hello packets.

$$
\begin{aligned}
&void\ TBR :: send\_hello(Packet * p)\\
&\{\\
&struct\ hdr\_TBR\_request\ *rq\ =\ HDR\_TBR\_REQUEST(p);\\
&TBR\_Neighbour\ *nb;\\
&nb\ =\ nb\_lookup(rq\ ->\ rq\_dst);\\
&\qquad p->msg\ =\ HDR\_TBR\_REQUEST("TRUST\ ECHO")\\
&\qquad if(nb\ ==\ 0)\\
&\qquad\qquad ERROR(No\_DST);\\
&\qquad send\_echo(nb,rq,p);\\
&\}
\end{aligned}
$$

Aside from identifying the neighbor and determining the hop count of the neighboring nodes, the routing protocol in the proposed system uses echo packets to share each node's trust value. These features are defined in the node's $send\_hello()$ and $recv\_hello()$ functions.cc file. Below the $send\_hello()$ and the $recv\_hello()$ function is depicted.

$$
\begin{aligned}
&void\ TBR::recv\_hello(Packet\ *p)\\
&\{\\
&struct\ hdr\_TBR\_reply\ *rp\ =\ HDR\_TBR\_REPLY(p);\\
&TBR\_Neighbour\ *nb;\\
&TBR\_tt\_entry\ tt;\\
&nb\ =\ nb\_lookup(rp->rp\_dst);\\
&p->msg\ =\ HDR\_TBR\_REQUEST("TRUST\_ECHO")\\
&if(nb\ ==\ 0)\\
&nb\_insert(rp->rp\_dst)\\
&Packet::free(p);\\
&Node\ *sender\_node\ =\ Node::get\_node\_by\_address(rp->rp\_dst);\\
&Node\ *\ receiver\_node\ =\ Node::get\_node\_by\_address(index);\\
&sender\_node->addNeighbour(receiver\_node);\\
&receiver\_node->addNeighbour(sender\_node);\\
&tt->t\_value\ =\ TBR\_Neighbour::get\_value(rp->rp\_tval);\\
&\}
\end{aligned}
$$

The $send\_hello()$ function calls the calculate trust() function, which computes the trust value using the equation above. The $calculate\_trust()$ function is shown in Figure 5. This function is also defined in the file node.cc.

$$void\ TBR::calculate\_trust()$$

```
{
TBR_tt_entry tt;
double W1, W2, W3, W4, W5, W6;
double rem_energy, drop_nos, fwd_nos, delay, cycle − time, reput;
W1 = −0.3;
W2 = 0.25;
W3 = −0.15;
W4 = 0.1;
W5 = −0.1;
W6 = −0.1;
rem_energy = tt−> get_battery_value();
drop_nos = Node::drop_packets();
fwd_nos = Node::forward_packets();
delay = Node::get_nodeDelay();
cycle_time = Node::get_SleepTime();
reput = tt → get_tvalue(Node :: get_node_by_address(Node::currentNode()));
tt−> tvalue = tt−
              > tvalue − ((W1 ∗ rem_energy) + (W2 ∗ drop_nos) + (W3 ∗ fwd_nos)
              + (W4 ∗ delay) + (W5 ∗ cycle_time) + (W6 ∗ reput));
}
```

When the hello packet is received, the trust value of other neighbor nodes is extracted and passed to the function *neighbor_list*(). Figure 6 depicts the *neighbour_list*() function. This function finds neighbors with trust values greater than the threshold and adds them to trusted neighbors' linked list data structure. The routing algorithm uses this data structure to determine the best path.

```
Node ∗ TBR_rt_entry TBR::neighbour_list()
{
TBR_tt_entry ∗tt,∗tt1;
for(tt = ttable.head(); tt! = ttable.end; tt = ttn)
for(tt1 = ttable.head(); tt1 ! = ttable.end; tt1 = ttn)
if(tt−> tvalue < tt1−> tvalue))
tt−> swap_entry(tt_tvalue, tt1−> tvalue);
ttn = tt1−> tt_link.next;
}
ttn = tt−> tt_link.next;
return (∗TBR_rt_entry) ttable.head;
}
```

In the aodv.cc file, a user-defined function *get_nexthop*() is defined, which retrieves the trusted neighbor's list from the trusted neighbor structure stored in *my_route.cc* file. The *get_nexthop*() function is illustrated in Figure 7. This trusted neighbor list is used to determine the best path. This function returns the node that will forward the packet.

```
Node ∗ TBR_nexthop TBR::get_nexthop()
{
TBR_tt_entry ∗tt;
for(tt = ttable.head(); tt! = ttable.end; tt = ttn)
```

$if(tt{-}>tvalue >= TBR::trust{-}>threshold))$
$tt{-}>nexthop\_entry(tt{-}>getAddress());$
$ttn = tt{-}>tt\_link.next;$
$return\ (*TBR\_nexthop)\ tt{-}>getAddress();$
$\}$

In route.cc, a user-defined function $route\_update()$ is defined, assigning the next-hop address for each packet to arrive at the destination. This function calls $get\_nexthop()$ to determine the next node to which a packet should be forwarded. Based on the $get\_nexthop()$ function, the routing algorithm forwards the packet to the neighbor while avoiding the compromised nodes.

Following the identification of the next hop, the node forwards the packet using simple cryptographic encryption. Because the trusted neighbor list generation phase occurs before the Routing, compromised nodes are avoided. All attacks from compromised nodes will be avoided as a result.

Furthermore, the trust-based routing scheme does not frequently broadcast echo packets, reducing energy consumption. This will improve the system's energy efficiency. The final encryption scheme improves data confidentiality

The energy profile of nodes can be activated in NS2, and the energy drain can be examined. The energy consumption in each case is calculated from the trace file to determine the weight of each parameter. This experiment is repeated several times with varying numbers of network nodes. For example, the experiment is conducted with 20, 40, 60, and 80 network nodes, and the corresponding energy utilization is recorded.

Table 3 shows the energy consumption of the parameters chosen in the proposed scheme. The total percentage does not add up to 100 because a few less critical parameters were ignored in the experiment.

Table 3: Energy Utilization of Network Parameters (Initial Energy: 100)

Parameters / No of Nodes	**20**	**40**	**60**	**80**	**Avg.**	**(%)**
Remaining Energy	29.67	31.09	30.45	30.01	30.30	30%
No. of Packets Forwarded	14.56	15.17	16.02	14.71	15.11	15%
Sleep Cycle Time	9.67	10.19	10.91	11.21	10.49	10%
No. of Packets Dropped	26.12	25.65	25.12	26.1	25.74	25%
Delay	8.98	10.45	10.16	9.96	9.88	10%

The proposed scheme considers the effect of network parameters on a node's trust and the effect of neighbor nodes. This parameter is known as reputation, and it is assigned the left-out percentage.

Finally, weights are assigned based on the tabulated data. The trust value will be computed based on the equation in Eq. [2], and this is accomplished through a user-defined function "*calculate_trust()*" included in the "*node.cc*" file.

The proposed scheme is unique because it computes a node's trust value using a weighted summation approach. Each of the parameters listed above has a weightage that influences the trust value computation. First, the weights are chosen experimentally. The computed trust value is then distributed to the neighbors so that the nodes can use it in the trust-based routing process. Once a node has received trust values from its neighbors, it can generate a trusted neighbor list used in the routing process.

The trusted neighbor's list is used by trust-based routing to find the best path through those nodes. The idea behind routing only on trusted neighbors is that compromised nodes will not be chosen as the next forwarding node.

Only when compared to other similar approaches, can the effectiveness of the proposed system be demonstrated. As a result, the proposed method's performance is compared to two parallel approaches. The first is the standard AODV routing scheme available in NS2. The second is a variant of the AODV routing protocol proposed in (Choi & Song, 2006). The authors of that modified AODV routing scheme modified AODV to use neighbors' trust value to decide the next hop. Each node computes a trust value based on a few parameters, then shares with its neighbors. As a result, the node chooses the neighbor with the highest trust value as the next-hop node. Three WSN network parameters are compared to analyze performance and validate the claim that the proposed system is energy efficient: throughput, packet delivery ratio, and remaining battery energy. The details are not provided because Delay, Packet Delivery Ratio, and Remaining Energy are standard network parameters.

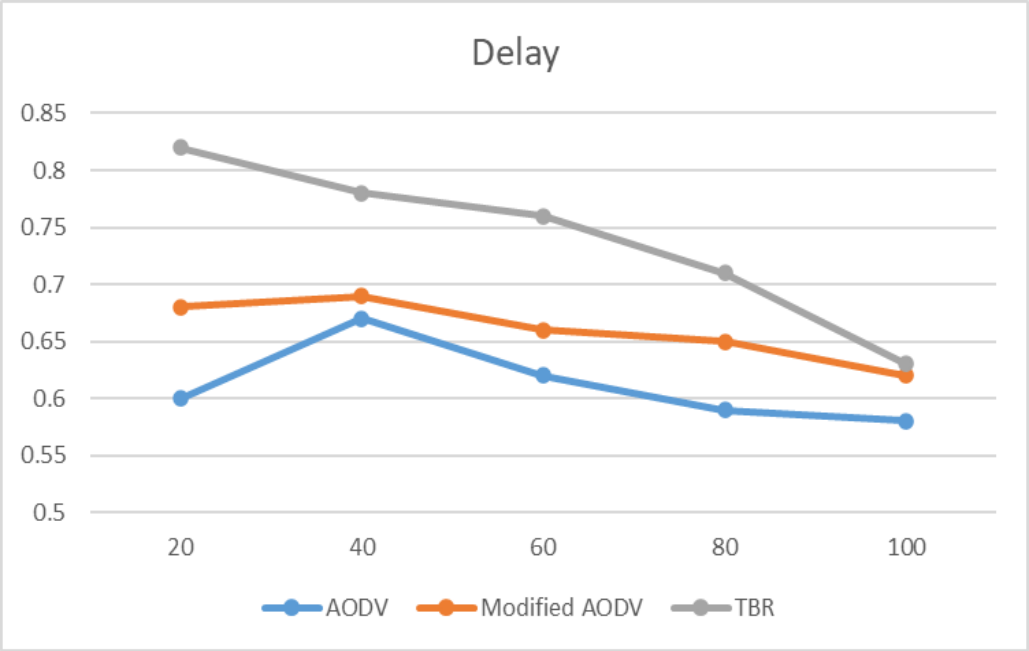

Figure 4: Delay Comparison

Figure 4 compares traditional AODV-implemented WSN, Cryptographic Security Scheme-based WSN, and the proposed system in terms of delay. The proposed system has a longer delay than the other two schemes, but it eventually improves and performs similarly to the other two schemes. The long initial delay is caused by the number of times nodes compute their trust value and gather the trust value of their neighbors.

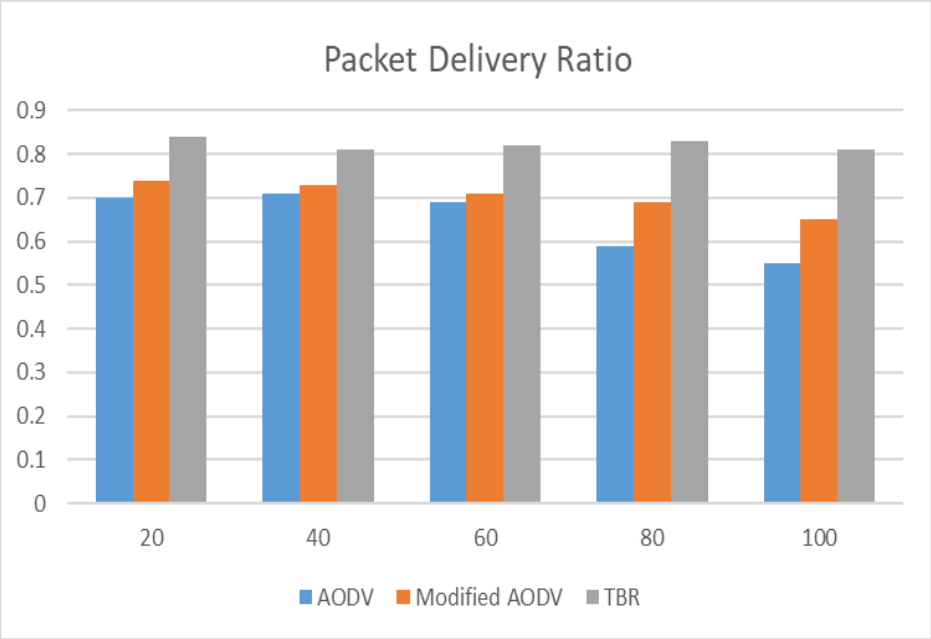

Figure 5: Packet Delivery Ratio Comparison

In Packet Delivery Ratio, Figure 5 compares traditional AODV-implemented WSN, Cryptographic Security Scheme-based WSN, and the proposed system. The proposed PDR improves over time and outperforms the other two methods.

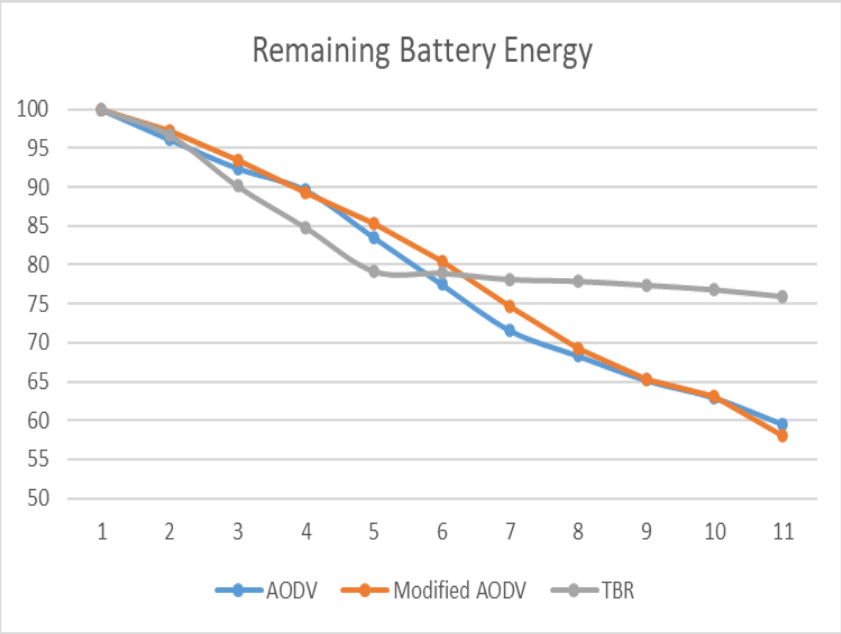

Figure 6: Remaining Battery Energy Comparison

Figure 6 compares traditional AODV-implemented WSN, Cryptographic Security Scheme-based WSN, and the proposed system in terms of remaining battery energy. Because the proposed method is computationally simple and consumes less power, the remaining battery energy of the proposed system

is always greater than that of the other two schemes. Furthermore, the trust-based Routing introduced is an energy-efficient routing technique because it broadcasts fewer echo packets.

Furthermore, we evaluate the network performance for varied number of malicious node and identify the packet delivery ratio. The obtained performance is compared with existing schemes such as SQEER (Kalidoss et al., 2020), ELPC (Dharini et al., 2020), QEBSR (Rathee et al., 2019), and ATRP (Khalid et al., 2019). Below given figure 7 depicts the overall packet delivery performance in terms of packet delivery rate.

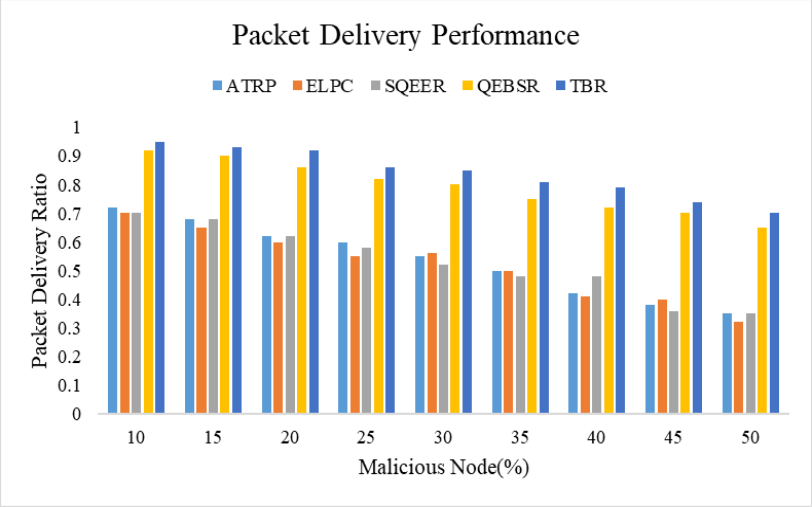

Figure 7: Packet Delivery Performance for Varied Number of Malicious Nodes

The average packet delivery performance is obtained as 53.5%, 52.1%, 53%, 79.1%, and 83.8% by using as ATRP, ELPC, SQEER, QEBSR, and TBR, respectively. The traditional schemes adopt the static trust management scheme whereas proposed approach updates the trust values every iteration. Moreover, proposed approach uses indirect trust management scheme.

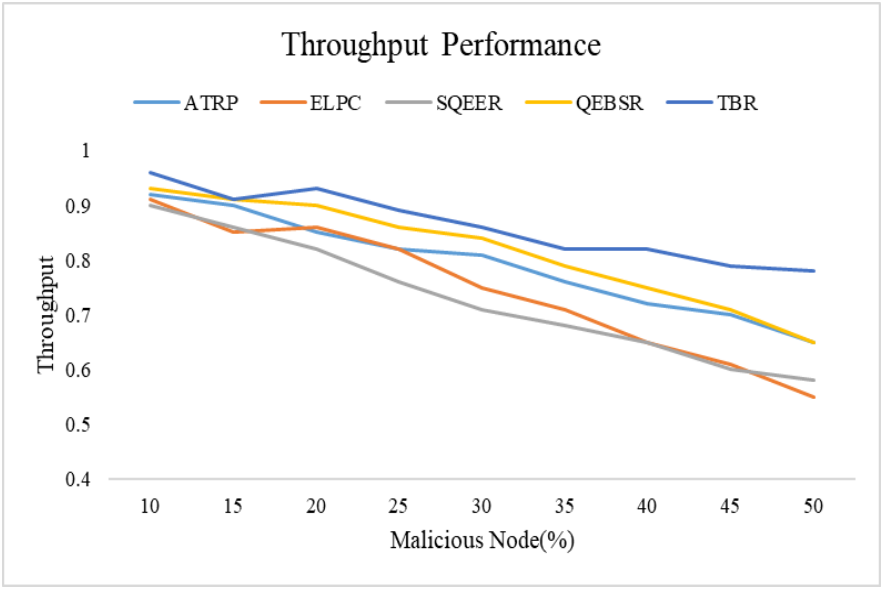

Figure 8: Throughput Performance for Varied Number of Malicious Nodes

Similarly, we measure the throughput performance, the comparative analysis is presented in figure 8. The average throughput performance is obtained as 0.79.22%, 74.56, 72.89%, 81.56%, and 86.22% by using ATRP, ELPC, SQEER, QEBSR, and TBR, respectively.

6 Conclusion

This paper proposes a novel data security scheme based on trust value computation to remove compromised nodes from decision-making and modify data transmitted across the WSN. The proposed scheme is a non-cryptographic scheme that employs a trust value for each node, with the trust value serving as the metric for identifying compromised nodes. Because the proposed scheme is non-cryptographic, it is less computationally complex than the most well-known cryptographic schemes used for data security, and thus it is energy efficient. The Network Simulator is used to simulate the proposed system (NS2). The proposed system considers several parameters when calculating trust values. It is a weighted approach, including more parameters and reviewing the weighted scheme of parameters in the trust value computation. The results show an increase in network lifetime by lowering the energy consumption while transmission of data in proposed security scheme.

References

[1] Al Shehri, W. (2017). A survey on security in wireless sensor networks. *International Journal of Network Security & Its Applications (IJNSA)*, *9*(1), 25-32.

[2] Awais, A., & Gouri, P. (2017). A Novel Composite Routing Strategy to Enhance Trust and Network Lifetime in WSN. *International Journal of Advanced Technology and Innovative Research*, *9*(7), 1224-1231.

[3] Bade, A. M., & Garba, A. A. (2019). A Review on Security Issues in Wireless Sensor Networks. *International Journal of Recent Academic Research*, *1*(5), 159-164.

[4] Chelli, K. (2015). Security issues in wireless sensor networks: Attacks and countermeasures. *In Proceedings of the World Congress on Engineering*, *1*(20), 876-3423.

[5] Choi, K. J., & Song, J. I. (2006). Investigation of feasible cryptographic algorithms for wireless sensor network. *In IEEE 8th International Conference Advanced Communication Technology*, *2*.

[6] Dharini, N., Duraipandian, N., & Katiravan, J. (2020). ELPC-trust framework for wireless sensor networks. *Wireless Personal Communications*, *113*, 1709-1742.

[7] Dhivvya, J. P., Jayakrishnan, V. M., Thomas, E. K., Ramesh, M. V., & Divya, P. (2017). Towards energy conservation in campus using Wireless Sensor Network. *In IEEE global humanitarian technology conference (GHTC)*, 1-6.

[8] Kalidoss, T., Rajasekaran, L., Kanagasabai, K., Sannasi, G., & Kannan, A. (2020). QoS aware trustbased routing algorithm for wireless sensor networks. *Wireless Personal Communications*, *110*(4), 1637-1658.

[9] Kasar, S., Khairnar, D. G., & Sharma, M. (2015). An energy saving routing mechanism for intrusion prevention in wireless sensor networks. *International Research Journal of Engineering and Technology*, *2*(4), 276-281.

[10] Khalid, N. A., Bai, Q., & Al-Anbuky, A. (2019). Adaptive trust-based routing protocol for large scale WSNs. *IEEE Access*, *7*, 143539-143549.

[11] Naveenraj, R., & Student, M. P. (2017). An Efficient Secure Data Transmission & False Detection in Wireless Sensor Networks. *International Journal of Engineering Science and Computing*, 2017.

[12] Oke, J. T., Agajo, J., Nuhu, B. K., Kolo, J. G., & Ajao, L. A. (2018). Two layers trust-based intrusion prevention system for wireless sensor networks. *Advances in Electrical and Telecommunication Engineering*, *1*, 23-29.

[13] Pawar, M., & Agarwal, J. (2017). A literature survey on security issues of WSN and different types of attacks in network. *Indian Journal of Computer Science and Engineering*, *8*(2), 80-83.

[14] Rathee, M., Kumar, S., Gandomi, A. H., Dilip, K., Balusamy, B., & Patan, R. (2019). Ant colony optimization based quality of service aware energy balancing secure routing algorithm for wireless sensor networks. *IEEE Transactions on Engineering Management*, *68*(1), 170-182.

[15] Simi, S., & Ramesh, M. V. (2019). Intelligence in wireless network routing through reinforcement learning. *International Journal of Communication Networks and Distributed Systems*, *23*(2), 231-251.

[16] Sreevidya, B., & Rajesh, M. (2017). Enhanced energy optimized cluster based on demand routing protocol for wireless sensor networks. *In International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016-2019.

[17] Sreevidya, B., & Rajesh, M. (2018). False data injection prevention in wireless sensor networks using node-level trust value computation. *In International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2107-2112.

[18] Sreevidya, B., Rajesh, M., & Mamatha, T. M. (2018). Design and development of an enhanced security scheme using RSA for preventing false data injection in wireless sensor networks. *In Ambient Communications and Computer Systems: RACCCS 2017*, 225-236.

[19] Zhong, H., Shao, L., & Cui, J. (2016). A lightweight and secure data authentication scheme with privacy preservation for wireless sensor networks. *In International Conference on Networking and Network Applications (NaNA)*, 210-217.

Authors Biography

B. Sreevidya, currently serves as Assistant Professor (Senior Grade) at Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, India. She is currently pursuing his Ph.D. She has been with Amrita University for more than 17 years now. She pursued his Master's in Computer Science and Engineering from Visveswaraiah Technological University, Bengaluru. Her area of interest includes Sensor Networks, Wireless Networks and Data Security. She has around 20 publications in the domain of wireless sensor networks and data mining.

Dr.M. Supriya, currently serves as Associate Professor at Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, India. She has been with Amrita University for more than 19 years now. Her area of interest includes Cloud Computing, Distributed Systems and Computer Security. She has around 30+ publications in the domain of cloud computing and distributed systems.