Trust-based Selfish Node Detection Mechanism using Beta Distribution in Wireless Sensor Network

Kanchana Devi V* & Ganesan R

School of Computing Science and Engineering, VIT Chennai, Vandalur - Kelambakkam Road Chennai, Tamil Nadu - 600 127, India
*E-mail: kanchanadevi@vit.ac.in

Abstract: Wireless sensor networks (WSNs) are placed in open environments for the collection of data and are vulnerable to external and internal attacks. The cryptographic mechanisms implemented so far, such as authorization and authentication, are used to restrict external sensor node attacks but cannot prevent internal node attacks. In order to evade internal attacks trust mechanisms are used. In trust mechanisms, firstly, the sensor nodes are monitored using the popular Watchdog mechanism. However, traditional trust models do not pay much attention to selective forwarding and consecutive packet dropping. Sometimes, sensitive data are dropped by internal attackers. This problem is addressed in our proposed model by detecting selective forwarding and consecutive failure of sending packets using the Beta probability density function model.

Keywords: *beta distribution mathematical model; consecutive failure; internal attacks; selfish nodes; wireless sensor network.*

1 Introduction

Some well-known security techniques that are practiced in wireless sensor networks are integrity, confidentiality, availability, and authentication. The most common attacks in WSN can be divided into two categories, namely internal attacks and external attacks. Cheng, *et al.* [1] state that external attacks are initiated by sensor nodes that do not belong to the network, whereas internal attacks are triggered by nodes of the network itself by dropping data or control packets, and alter or misroute data packets. Usually, cryptographic mechanisms are used to prevent external attacks. However, these cannot completely prevent internal attacks, which can cause major problems in life-critical applications like robotic surgery, autonomous vehicles, etc.

Trust helps to overcome several problems that occur in unattended open heterogeneous environments. Trust management works with several components combined together to calculate the trust of a particular sensor node:

1. Collecting information by monitoring

Received December 8th, 2018, Revised March 23rd, 2019, Accepted for publication April 10th, 2019.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2019.13.1.6

2. Calculating the trust value based on the collected information
3. Decision-making based on the calculated trust value
4. Updating the information to all other nodes

Every sensor node in the network is watched by the well-known Watchdog mechanism, tracking certain node parameters, such as packet forwarding count and packet dropping count towards the sink node.

The Entropy trust model and the Bayesian trust model are two well-known trust models used to calculate the trust of sensor nodes by means of the information collected by the Watchdog mechanism, as specified by Che, *et al.* [2]. Then, the calculated trust value is compared with an estimated threshold to be able to identify internal attacks. When selfish node behavior is detected, every node first checks the trust value of its neighboring nodes to find a trusted path to use for sending further packets.

In the presence of environmental changes, a WSN as depicted in Figure 1 is prone to dropping packets. This is convenient for attackers who aim to drop packets intentionally. In the case of such internal attacks it is hard for any defending mechanism to determine whether the packets were dropped because of environmental conditions such as noise, contention, or an internal attacker. Suppose packets were dropped by an internal attacker only for a short amount of time. When the node behaves normally again, this activity cannot be classified as trusted or untrusted by traditional trust mechanisms. In this kind of scenario it is difficult to fully protect against internal attacks.

Ad-hoc networks, eCommerce feedback and peer-to-peer networks are some of the network environments where trust plays a vital role. The Beta trust model gives good results in such networks, where the trust of each node is adjusted by considering direct information and indirect (second-hand/recommendation) information. This indirect information can be used for faster manipulation of the trust values of the nodes in a mobile node environment.

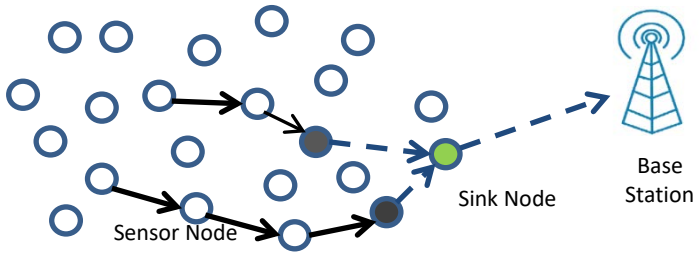

Figure 1 Structure of wireless sensor networks.

2 Related Work

Marti, *et al.* [3] presented mitigation of routing misbehavior in MANET, examining two new techniques, Watchdog and Path Rater, to identify changing behavior in ad-hoc nodes. They achieved good improvement in efficiency and accuracy by optimizing various parameters such as overhead in transmission. In the experiment, misbehaving nodes were identified using the Dynamic Source Routing Protocol. Varshney, *et al.* [4] implemented the Watchdog protocol over Adhoc On-Demand Distance Vector (AODV) in a mobile ad-hoc network, mainly to identify black hole attacks in MANET. Their method, called Watchdog-AODV, establishes a new route to send further packets. The simulation results showed improvisation in packet delivery, throughput and routing overhead.

Ammendola, *et al.* [5] have proposed a hierarchical watchdog method for finding systemic faults in a distributed environment. This mechanism can rapidly identify fake nodes by visualizing the global state of the system. The proposed method fixes single-point failures in distributed environments as a solution for the fake node problem through double diagnostic messages (systemic resilience). Liu, *et al.* [6] have demonstrated reliability oriented transmission in WSNs, addressing problems such as low success rates and poor energy efficiency in a scalable network. They developed a proliferation routing scheme that combines three different components, i.e. random disperse, reproduction and a path rater scheme that depends on the capability of the network. The results showed good improvisation performance (around 80%) with a hop-based routing protocol.

Shen, *et al.* [7] have proposed an evolutionary game-based theoretical approach. They used game theory to develop a method for making decisions to check whether a node can be trusted or not. Incentives are provided for the trusted nodes. A simulation showed stability and enhanced security for the network using this approach. Sun, *et al.* [8] introduced a novel trust aware routing protocol for WSNs and examined multiple attributes of each sensor node towards energy, recommendation, data and communication. The authors integrated routing and maintenance through a sliding time window scheme in a scalable environment. The experimental results revealed 19% improvement in packet delivery and 11% improvisation in time consumption in routing.

Ishmanov, *et al.* [9] have proposed a new trust mechanism for secure routing in WSNs based on energy consumption and attack resiliency. They analyzed various trust-aware routing protocols to find the best solution. The mechanism consists of three components to overcome the shortcomings of existing models: (i) monitoring and learning, (ii) trust evaluation, and (iii) recommendation

management. Meng, *et al.* [10] proposed a Bayesian interference based system to defend against insider attacks, focusing on leakage of sensitive information by misbehaving nodes. The efficiency of the proposed model in identifying malicious sensor nodes in a real-world environment was demonstrated.

Alsaedi, *et al.* [11] have proposed a mechanism based on energy trust in a clustered WSN to identify Sybil attacks. It minimizes the communication overhead with a data aggregation policy. The results showed that this method has an efficient and robust detection ratio in identifying Sybil attacks. Also, this method removes the exchange of feedback between sensor nodes. Li W, *et al.* [12] and Lin [13] have developed a system to prevent complex attacks and betrayal attacks by enhancing identification accuracy. The authors designed a sensitivity aware intrusion detection system. In the experiment, various classifications were compared to examine the correctness of the proposed model.

3 Packet Transfer and Attack Types

The data sensed from the environment are transmitted to a sink node through single-hop or multi-hop communication. In single-hop communication, packets are usually directly transmitted to a sink node in one hop. However, sink nodes must be placed within the range of the transmitting node and in this type of communication the energy required for transmission is quite high. As described in Farooq, *et al.* [14], multi-hop communication is better suited for energy constrained devices. In multi-hop communication, the sensor node depends on a neighboring node for transmitting the packets to the sink node. Before forwarding a packet, the corresponding trust value of each node is checked. For collecting the evidence to compute the trust value, every node is equipped with a Watchdog mechanism. Marti, *et al.* [3] proposed a basic Watchdog mechanism in a dynamic sourcing routing protocol for nodes to monitor each other to check whether transferred packets are forwarded by the neighboring nodes.

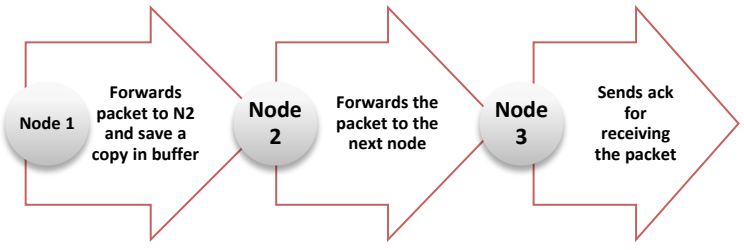

Figure 2 Packet forwarding and monitoring process.

Referring to Figure 2, assume that Node 1 forwards a packet to Node 2, where every node is equipped with the Watchdog mechanism. After transmitting the packet to Node 2, Node 1 keeps a copy of it in its buffer for the purpose of retransmission and conformity checking. As soon as packet delivery at Node 3 is confirmed, Node 1 discards the packet from its buffer and counts it as successful transmission. Node 1 senses this packet forwarding since Node 3 is within the range of Node 1. If time-out occurs, the transaction is counted as failed transmission.

Packet dropping attacks are categorized into three types: black hole attacks, which drop both control packets (routing packets) and data packets (payload); the second type are gray-hole attacks, which drop only certain packets; and the third type are on-off behavior attacks, which show dynamic behavior. These attacks are difficult to detect because it is hard to distinguish whether the packets are dropped by an attacker or because of heavy traffic in the network (network congestion).

4 Preventive Mechanisms

Every node in the network works in promiscuous mode, probing the other nodes in the network for security purposes. Figure 3 shows the node-to-node communication within the network, the node communication range, and malicious nodes. The trust value is manipulated based on the number of successful transactions and the number of failed transactions. Later, the trust value is calculated by another component of the mechanism. The packet forwarding mechanism is monitored by the Watchdog mechanism and the evidence is collected as successful and failed transmission counter values. Later, these counter values are used by another component of the mechanism to detect internal attackers.

Various trust modules and models have been used in previous studies to manipulate the trustworthiness of each node in a network. Some of the trust models used are: Beta model, fuzzy model, game-based approach, Bayesian trust model, and Entropy trust model. Among these models, the Beta model has some weaknesses toward internal packet dropping since it fails to address continuous or consecutive failures (continuous dropping of packets). This consecutive dropping of packets is detected by giving a penalty if the number of dropped packets falls under a given threshold. Also, if this consecutive dropping of packets is launched after gaining maximum trust value, this poses a very serious problem, leading to network partition. Meanwhile, packet droppers can only be detected only after dropping a certain number of packets, depending on a threshold value. This is taken as the core problem and is handled by preprocessing the collected evidence in order to detect packet droppers earlier.

The algorithm for preprocessing the evidence collected from the Watchdog mechanism is given below.

1. Begin
2. $N \leftarrow Source\ Node$
3. $node_1, node_2, \ldots\ldots, node_m$ are the neighboring nodes of N
4. // Trust T is calculated every 60 seconds
5. // get the data from the watchdog regarding the successful transmission every 20 seconds
6. $//failure_t \rightarrow Cumulative\ failure$ $//success_t \rightarrow Cumulative\ success$
7. $for\ every\ session\ between\ N\ and\ Node_i,\ where\ 1 \leq i \leq m$
8. $func\ Evidence\_calc(Success,\ Failure)$
9. { $for\ every\ T\ secs${
10. $for\ every\ t\ secs$ {
11. $success_t = success_t + success$
12. $check(failure > Th_2 \&\& failure < Th_1)$ //$Th_1 > Th_2$
13. $failure_t = \left(\frac{1}{4} \times failure_t\right) + failure$ //penalty
14. $otherwise$
15. $failure_t = failure_t + failure$ } } }
16. End

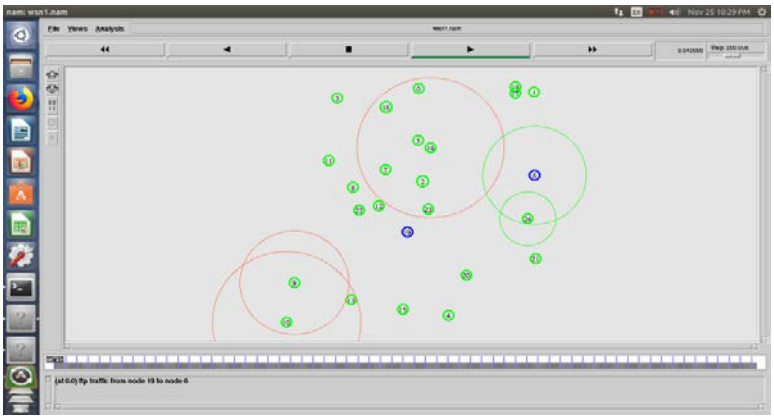

Figure 3 Screenshot of network simulator (network setup).

4.1 Proposed Beta Probabilistic Density Function-based Trust Model

This Beta probabilistic density function-based trust model depends on evidence collected by the Watchdog mechanism by counting the successful and failed transactions in promiscuous mode. Eq. (1) shows the formula for the initial trust value in each sensor node.

$$Trust\ Value\ T(node) = \frac{Success+1}{Success+Failure+2} \quad (1)$$

Usually, the trust value of the sensor nodes using the above equation will be between 0 and 1. Positive 1 is added to the numerator part and a positive 2 is added to the denominator part to convey the assumption that in order to assess the trust value of a particular sensor node, at least two transactions should take place. This equation is reliant upon the Laplace law. Thus, every sensor node is initialized with trust value 0.5. However, every sensor node is monitored in promiscuous mode and updated often. Instead of transmitting every updated value, the current updated value can be transmitted every T seconds to overcome the problem of bottleneck and overhead in low-power devices such as the nodes in a WSN. Figure 4 shows the network setup in Network Simulator 2 (NS2).

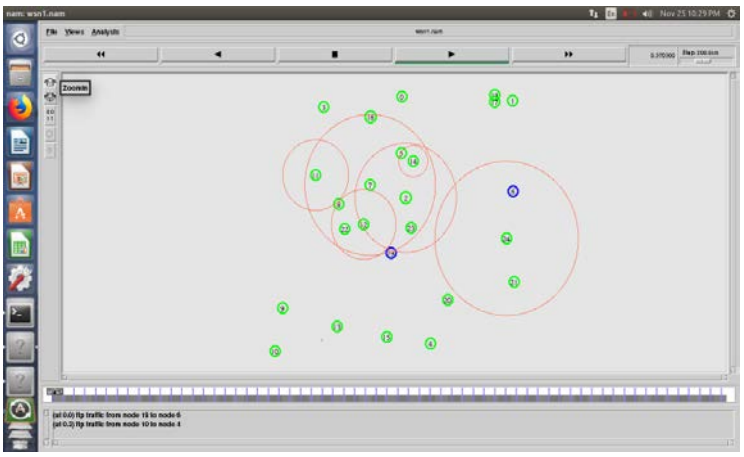

Figure 4 Screenshot of network shows the nodes that are not within range.

$Success_t$ represents the total number of cumulative successful transactions and $Failure_t$ represents the total number of cumulative failed transactions at T seconds.

$$Success_{new} = Success_{old} + Success_t \quad (2)$$

$$Failure_{new} = Failure_{old} + Failure_t \quad (3)$$

The newly observed data are stored in the sensor node, where space for holding new values may not be supported in all cases. A separate solution has to be found for storing the old data. The 'history factor' concept is introduced to overcome this problem. Here, β gives the updated trust value from each sensor node's history. Usually the value of β lies between 0 and 1.

$$Success_{old} = \sum_{T=1}^{k} Success_T \times \beta^{(k-T)} \quad (4)$$

$$Failure_{old} = \sum_{T=1}^{k} Failure_T \times \beta^{(k-T)} \quad (5)$$

4.2 Decision-making based on Trust Value

Decisions are made by means of the above trust value through Watchdog mechanism and the Beta probabilistic density function. Separate counters for both success and failure are set and captured with a trust variable using the Beta probabilistic density function to check with a specified threshold value. The threshold is represented by θ, which is compared with the manipulated trust value before transmitting any packet. If the trust value is greater than the threshold value, then the packet can be transmitted since this node is considered cooperative and non-selfish, i.e. it is trusted. If the trust value is smaller than the threshold value, then the corresponding node is marked as a selfish node. Further transactions are not allowed with this particular node. Instead, the packets are rerouted to a neighboring node using the routing protocol.

4.3 Updating the Information to Other Nodes

Once a node has been categorized as selfish or malicious, the afflicted node disconnects the link and triggers the routing process to find a disjoint route. Also, the detection of the selfish node is communicated to both the source node and the sink via a disjoint route.

5 Energy Consumed by Beta Density Function Model

In the proposed Beta density function model, all sensors are observed and their trust values are manipulated. The core module (microprocessor) consumes some energy for transferring and retrieving the observed data to and from the memory module and some power is utilized by the analog to digital converter. Usually, most of the power is used for radio communication. Ahmed, *et al.* [15] emphasize that the power used for extra operations, incorporated along with the already existing modules, also needs to be considered. Protocols should be carefully designed in such a way that embedded operations do not consume extra power. In the proposed model, an energy model is implanted along with the configuration of the nodes. In most of today's sensors, the energy is renewed with the help of solar cells.

Figure 5 shows one of the output files (trace file) of the NS-2 simulator, where the energy consumed by each node before and after transmission has been captured. Malicious nodes are automatically disconnected from the route, depending on their trust value. The experimental results showed that less energy is used to calculate the trust value and to disconnect untrusted nodes from the

routing path. Figure 6 shows various components in the sensor nodes, i.e. the power unit, the memory unit and the processing unit.

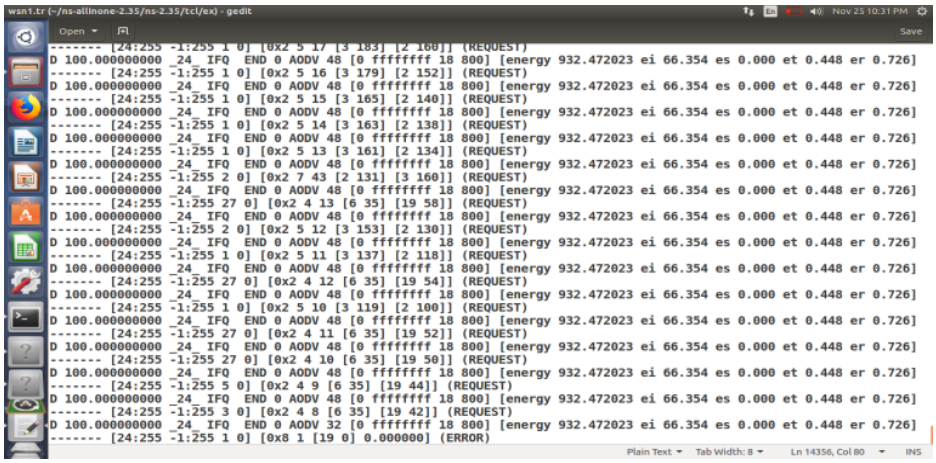

Figure 5 Screenshot of trace file in NS2 showing the trust calculation in each node.

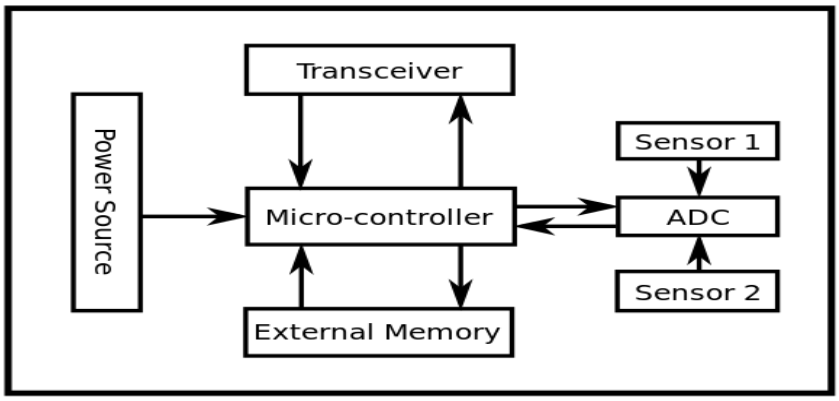

Figure 6 Sensor architecture.

5.1 Pros and Cons of the Beta Density Function Model

This section examines the pros and cons of the Beta density function model. Pros: it is a very flexible mathematical model and it is simple to use since it is statistical. It uses both the behavior and the feedback of the sensor nodes. Cons: the Beta probabilistic density function suffers from heterogeneity, indicated by using two parameters, α and β. It is easier to use a single parameter instead of two. Certain security issues associated with this preprocessing module, need to be addressed, for example collision and power constraints.

5.2 Issues in the detection phase

Deciding the threshold value demands more attention. Generally, the threshold value is tailored by the application. If the network handles sensitive data, then the threshold value ought to be set to 0.95. Hence, if a sensor node drops 10 packets out of 100 packets, the node is considered untrusted and further packets are rerouted.

If the application is not too sensitive or a safety-critical application, then the threshold value can be varied between 0.5 and 0.7. The total number of packets dropped by each sensor node is calculated using Eq. (7).

$$T_{total} = \frac{[Success_{old}+1-(\theta*(Success_{old}+2))]}{\theta} \quad (7)$$

However, the ultimate aim is to reduce the total number (T_{total}) of the drop in packets. This T_{total} is used to maintain or update the θ value, because over a period of time an internal attacker may gain control or predict the threshold θ value. This control takeover is enough to turn a normal sensor node into a selfish node, which is prevented by varying θ depending on the number of transactions that this particular sensor node makes.

If a sensor node drops a packet, it is not a normal node, because in a sensor network the nodes are supposed to forward all of the packets they receive. Thus, this node could be under control of an internal attack. Another cause could be a faulty node. As mentioned by Chen *et al.* in [16], if there is continuous dropping of packets by a node, the trust value gets automatically updated in order to punish it and reduce its functionality by decrementing the trust level, as given by Eq. (8).

$$Success_{new} = Success_{old} * \beta + Success_t \quad (8)$$

$$Failure_{new} = Failure_{old} * (1 - \beta) + Failure_t \quad (9)$$

Here, the value of variable β lies between 0 and 0.45. The maximum trust value is 0.75. The focus of this study was on identification of selfish nodes rather than increasing its trust value.

5.3 Algorithm for Punishing Continuous Packet Droppers

1. Begin
2. Initialize $Failure_p$ as 1
3. Initialize trust update value β to 0.9
4. Calculate the latest trust value
5. $Trust_{latest} = \frac{Success_t}{Success_t+Failure_t}$

6. Check whether the calculated $Trust_{latest}$ is smaller than threshold value θ. Then do the following
7. $Failure_p = Failure_p * 3$;
8. $Success_{old} = Success_{old}/Failure_p$;
9. Check if $Success_{old}$ is less than 0.1, then $Failure_p = 1$;
10. $Success_{new} = Success_{old} * \beta + Success_t$
11. $Failure_{new} = Failure_{old} * (1 - \beta) + Failure_t$
12. Calculate the total overall trust using

$$Trust_{total} = \frac{Success_{new}+1}{Success_{new}+Failure_{new}+2}$$

13. Calculate the total overall trust with a penalty

$$Trust_{total} = \frac{Success_{new}+3}{Success_{new}+Failure_{new}+4}$$

The proposed model was experimentally tested using NS2. The number of sensor nodes varied from 26 to 60 nodes in order to check the correctness of the proposed model. The proposed model was compared with other recent models to prove its efficiency.

In the simulation setup, total number of sensor nodes was varied and the corresponding sink nodes were established. In this scenario, every sensor node in the network is equipped with the Watchdog mechanism, mainly to keep watch of the neighboring nodes by enabling promiscuous mode. Two kinds of attacks were simulated to check how our mechanism performed with 26 nodes and 60 nodes. Some of the nodes were set as selfish nodes that drop all incoming packets, thus launching a black hole attack as described by Otoum, *et al.* [17]. Another selfish node dropped only certain packets and showed dynamic behavior. This type of packet dropping is known as selective forwarding or a gray-hole attack.

Various trust updating models proposed by Che, *et al.* [2] were compared with the proposed model. The proposed model provides a simple and flexible solution and improves the lifetime of the network. Also, throughput is increased in the presence of a malicious node. Rerouting can be done by avoiding such nodes using the proposed trust mechanism.

Table 1 Trust value variation with time variation (seconds).

Time (in seconds)	Trust Value of Proposed Beta Model (Modified)	Bayesian Model	Entropy Model
500	0.9396	0.9676	0.9558
1000	0.4545	0.6601	0.8608
1500	0.2030	0.5142	0.6298

In Table 1, the trust value of a particular selfish node can be seen falling below the threshold value (0.5) with the proposed method, while with the other two models it still stayed above the threshold value.

6 Conclusion

Consecutive failure of packet sending (continuous packet dropping) is not addressed in any of the available trust models. Our proposed model is simple to implement and effective in safety-critical systems because it can find selfish sensor nodes irrespective of their dynamic behavior and reroute packets through a disjoint route. Our experimental result showed an improvement of efficiency compared with the original Beta reputation model and the Entropy trust model.

References

[1] Cheng, X., Luo, Y. & Gui, Q., *Research on Trust Management Model of Wireless Sensor Networks,* in IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 1397-1400, 2018.

[2] Che, S., Feng, R., Liang, X. & Wang, X., *A Lightweight Trust Management Based on Bayesian and Entropy for Wireless Sensor Networks,* Security and Communication Networks, **8**(2), pp. 168-175, 2015.

[3] Marti, S., Giuli, T.J., Lai, K. & Baker, M., *Mitigating Routing Misbehavior In Mobile Ad Hoc Networks,* in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp. 255-265, 2000.

[4] Varshney, T., Sharma, T. & Sharma, P., *Implementation of Watchdog Protocol with AODV in Mobile Ad hoc Network*, IEEE Fourth International Conference on on Communication Systems and Network Technologies, pp. 217-221. 2014.

[5] Ammendola, R., Biagioni, A., Frezza, O., Cicero, F.L., Lonardo, A., Paolucci, P.S., Rossetti, D., Simula, F., Tosoratto, L. & Vicini, P., *A Hierarchical Watchdog Mechanism for Systemic Fault Awareness on Distributed Systems,* Future Generation Computer Systems, **53**, pp. 90-99, 2015.

[6] Liu, Y., Zhu, Y., Ni, L. & Xue, G., *A Reliability-oriented Transmission Service in Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, **22**(12), pp. 2100-2107, 2011.

[7] Shen, S., Huang, L., Fan, E., Hu, K., Liu, J. & Cao, Q., *Trust Dynamics in WSNs: An Evolutionary Game-Theoretic Approach,* Journal of Sensors, Article ID 4254701, 10 pages, 2016.

[8] Sun, B. & Li, D., *A Comprehensive Trust-aware Routing Protocol with Multi-Attributes for WSNs*, IEEE Access, 6, pp. 4725-4741, 2018.

[9] Ishmanov, F. & Bin Zikria, Y., *Trust Mechanisms to Secure Routing in Wireless Sensor Networks: Current State of the Research and Open Research Issues*, Journal of Sensors, 2017.

[10] Meng, W., Li, W., Xiang, Y. & Choo, K.K.R., *A Bayesian Inference-Based Detection Mechanism To Defend Medical Smartphone Networks Against Insider Attacks*, Journal of Network and Computer Applications, **78**, pp. 162-169, 2017.

[11] Alsaedi, N., Hashim, F., Sali, A. & Rokhani, F.Z., *Detecting Sybil Attacks in Clustered Wireless Sensor Networks based on Energy Trust System (ETS)*, Computer Communications, **110**, pp. 75-82, 2017.

[12] Li, W., Meng, W., Kwok, L.F. & Horace, H.S., *Enhancing Collaborative Intrusion Detection Networks Against Insider Attacks Using Supervised Intrusion Sensitivity-based Trust Management Model*, Journal of Network and Computer Applications, **77**, pp. 135-145, 2017.

[13] Lin, K., Xu, T., Song, J., Qian, Y. & Sun, Y., *Node Scheduling for All-Directional Intrusion Detection in SDR-Based 3D WSNs*, IEEE Sensors Journal, **16**(20), pp. 7332-7341, 2016.

[14] Farooq, M.U., Wang, X., Yasrab, R. & Qaisar, S., *Energy Preserving Detection Model for Collaborative Black Hole Attacks in Wireless Sensor Networks*, in 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), pp. 395-399, 2016.

[15] Ahmed, A., Bakar, K.A., Channa, M.I., Khan, A.W. & Haseeb, K., *Energy-Aware And Secure Routing With Trust For Disaster Response Wireless Sensor Network*, Peer-to-Peer Networking and Applications, **10**(1), pp. 216-237, 2017.

[16] Chen, H., Wu, H., Zhou, X. & Gao, C., *Agent-based Trust Model in Wireless Sensor Networks*, IEEE Networking, and Parallel/Distributed Computing, pp. 119-124, 2007.

[17] Otoum, S., Kantarci, B. & Mouftah, H.T., *Hierarchical Trust-based Black-hole Detection in WSN-based Smart Grid Monitoring*, in IEEE International Conference on Communications (ICC), pp. 1-6, 2017.