Enhanced multi-attribute trust protocol for malicious node detection in wireless sensor networks

V RAM PRABHA^{1,*} and P LATHA²

¹V V College of Engineering, Tisaiyanvilai, Tirunelveli, India

²Department of Computer Science and Engineering, Government College of Engineering, Tirunelveli, India
e-mail: prabhasuresh.v@gmail.com

MS received 1 March 2016; revised 21 June 2016; accepted 18 September 2016

Abstract. There has been a tremendous growth in the field of wireless sensor networks (WSNs) in recent years, which is reflected in various applications. As the use of WSN applications increases, providing security to WSNs becomes a leading issue. This is complex due to the unique features of WSNs. This paper proposes a trust-based intrusion detection that uses multi-attribute trust metrics to improve detection accuracy. It uses an enhanced distributive trust calculation algorithm that involves monitoring neighbouring nodes and trust calculation using the trust metrics message success rate (MSR), elapsed time at node (ETN), correctness (CS) and fairness (FS). In addition to the normal communication-based trust property MSR, this paper uses effective parameters like ETN, which focuses on data and address modification attacks in an effective manner, and two social-interaction-based parameters CS and FS, which address trust-related attacks effectively. Simulation results show that the proposed method has higher performance and provides more security in terms of detection accuracy and false alarm rate.

Keywords. Wireless sensor network; intrusion detection; trust management; detection accuracy; false positive.

1. Introduction

Wireless sensor network (WSN) is a non-infrastructure network that is formed by easy and cheap deployment features with limited resources [1]. The main function of WSN is gathering data from environments and forwarding them to the Base Station (BS). Due to rapid development in science and technology, WSNs have been used in many fields [2]. For efficient energy utilization, cluster-based architectures are generally used [2, 3]. In the proposed work, a cluster-based architecture is used in order to reduce energy consumption. Cluster heads (CHs) need higher energy than sensor nodes (SNs) [1]. The architecture of cluster-based WSN is shown in figure 1.

As WSN is purely data-centric in nature, it is vulnerable to various attacks [4, 5]. To defend against these attacks, prevention-based and detection-based methods can be used [6]. Prevention-based techniques (e.g., cryptography, authentication protocols and access controls) rely on centralized infrastructure, and so when it is destroyed, the whole network may be affected. Even though the prevention-based methods prevent malicious behaviour to a particular level, there are chances for misbehaviour. Hence, second-level detection-based approaches are needed [7]. In

this work, an intrusion detection system (IDS) based on trust management is proposed for providing security to WSNs. IDS is a method used to detect the malicious behaviour of nodes, which disturbs the normal functioning of the networks. The main aim of IDS is to monitor the network continuously and to detect the possible intrusions in that network. This alerts the users when it detects intrusions. IDS can be classified into two types: anomaly detection and misuse detection [2]. In anomaly detection, the behaviour of the SNs is monitored and high discrepancy in the normal behaviour is noted to identify them as malicious nodes [8]. This can identify the new attacks also, but the false alarm rate is high. In misuse detection, the signatures of well-known attacks are stored in the database and the monitored behaviours are compared with that database to identify the malicious nodes. In this method, only the well-known attacks are identified and no new attacks can be detected. Hence, anomaly detection with high detection accuracy becomes an interesting research area [8]. Thus in this work, anomaly-based intrusion detection is considered.

Some trust-based schemes have been modelled recently to detect malicious nodes in WSNs. Nowadays, many researchers work on trust schemes for providing security in WSN, and it remains a challenging issue [9, 10]. The evaluation of trust is directly calculated from the past experiences or indirectly calculated by the

*For correspondence

Figure 1. Cluster-based WSN architecture.

recommendations given by others. But limited energy, limited memory capacity and open wireless medium lead to the necessity of an efficient trust framework for WSNs [11]. Since WSNs are data-centric in nature, attacks on WSN are aimed to either drop the packets that make the data unreachable to BS or modify the data and the address to make the information given to BS invalid [11]. Hence, this work aims at detecting these types of attacks, which are of greater importance.

The rest of the paper is organized as follows. A review of related work is presented in section 2. A brief overview of the proposed method is given in section 3. Section 4 describes the proposed trust calculation algorithm based on multi-attribute trust metrics. Section 5 deals with the results and their performance evaluation. Conclusion is given in section 6.

2. Related work

Very few schemes have been proposed for trust-based IDSs. In [12] a reputation-based trust management scheme was proposed in which each SN holds the information related to the reputation of the other neighbouring nodes in the network. They calculated the trust using Bayesian model beta distribution. But in this scheme, the nodes are not allowed to forward negative reputation information. An agent-based trust and reputation management (ATRM) scheme for WSNs was proposed in [9]. This distributed method calculates the trust of SNs in a clustered WSN based on agent-based scheme. In this method, generation and initiation of mobile agents were done by a single trusted authority. Due to this, it leads to a single point of failure. Parameterized and Localized trUst management Scheme (PLUS) for WSN was proposed in [10]. In their work they implemented a localized and distributed scheme that calculates both direct and indirect trust values. This scheme adds the hashed sequence number (HSN) to the control packets of the BS. This increases the size of

packets and, in turn, increases the computation cost also. A group-based trust management protocol [13] calculates the trust of a group of SNs. This reduces the memory usage as well as the cost of trust calculation. But it considers the trust as a value representing the whole group and message success rate (MSR) as the only factor for trust calculation, which makes this scheme an unreliable one. In [14] a trust model AEMP (addition encouragement, multiplication punishment) with two models was proposed. The first one is the basic model, where the direct values of trusts are calculated by the SNs and these values are not given as recommendations for the other nodes. In the second model called Integration Model, both the direct and the indirect values are considered, and the trust values are calculated by using the weighted sum approach.

In hierarchical trust management protocol [15], social attribute trust values are taken for calculating the overall trust value in addition to QoS parameters. In this method, trust calculation is done at two levels. One is at the SN level and the other at CH level. The authors have considered both the QoS and social trust attributes. They have taken the same energy and forwarding capabilities of a node as QoS trust attributes and the behaviour of nodes as social trust parameters. In our previous work [16], intimacy, honesty, energy and unselfishness were taken as trust parameters. However, in this work we have taken completely new trust attributes: elapsed time as QoS attribute and the correctness (CS) and fairness (FS) as social attributes along with MSR. This takes communication behaviour trust (CBT) and social behaviour trust (SBT) into account for finding whether a node is malicious or not. These efficient parameters are used in order to cover maximum possible attacks in the network. Time elapsed between the reception and the forwarding of a packet is considered as a communication behaviour parameter, which is an efficient one to address attacks that are aimed at modifying data or address in the packet. This is an efficient and simple one that works better than the integrity checks, which need more computation and time.

3. System model and assumptions

3.1 Trust definitions and metrics

Trust is generally considered as a belief on some entity based on behaviours and has different meanings when it is used in different applications [1, 7]. In the field of WSNs, this can be considered as a belief that a SN has on other nodes based on their past behaviours or interactions and also the recommendations or suggestions given by the other nodes [7, 11]. The nature of deployment of WSN leads to different attacks. Since it depends upon the connectivity and trustworthiness of the SNs, trust establishment is more important to improve the security of WSN. Recently, the concept of trust was applied to improve security [17–19],

Table 1. Trust metrics and the attacks identified.

Trust metric	Attacks identified
Message success rate	Routing attacks
Elapsed time at node	Data and address modification attacks
Fairness	Compromised and selfish node, trust-related attacks
Correctness	Compromised and selfish node, trust-related attacks

and the application of trust in WSN is an open issue. Selection of trust properties is application-driven. Since trust relationship has heavy subjectivity as well as uncertainty, a correct and applicable trust metric should be taken for establishing the degree of trust in a node. Multiple factors can be used to calculate the trust value more efficiently than a single value [15]. Hence, this work takes multiple attribute metrics and each metric is used to detect the specific type of attack. They are given in table 1 along with the attacks they address.

Here, four trust metrics that are different from those of [16] are taken to determine the trustworthiness of the nodes and they come under two types of trust values. The first one is the CBT, and the second one is the SBT. CBT value is taken due to the data-centric nature of WSN for addressing the issues related to routing [20], and SBT is taken to address the various attacks aimed at the trust protocol. This trust model calculates the trust values based on multi-attribute trust metrics, which efficiently identify the malicious nodes. MSR and interval time between reception and forwarding of the message are taken as QoS parameters. Unlike the previous trust management schemes in the literature, this work considers interval time as a parameter to address the data and the address modification attacks in an effective manner. To the best of our knowledge, this is the first work that considers the CS in the information given by a node and the FS in providing the information by a node as social trust parameters.

3.2 System model

We consider clustered wireless sensor network architecture for energy consumption with the assumption that a huge number of SNs are deployed in hostile environments. Each SN has its own ID. Nodes communicate with the other nodes in the network through multi-hop routing. The nodes are organized into clusters with the assumption that CHs are trusted nodes. Each node calculates the trust of its neighbours by monitoring its behaviour. For one-hop neighbours, the node calculates the trust based on direct observation. If they are not one-hop neighbours, then it gets the suggestions from their neighbours as indirect trust values and aggregates the trust values based on both direct and indirect

trust and then decides whether that node is malicious or not. Trust values are considered as a numerical value between 0 and 1. Initially, all the nodes are assumed to be trustworthy and assigned a highest trust value 1. For the first cycle, the trust value is fixed based on their reputation. The reputation value N_{rep} is evaluated based on the behaviour of those nodes in the previous cycle. Let N_r be the number of received packets and N_f be the number of packets forwarded by a node N. Reputation of node N is evaluated by the ratio of the number of packets forwarded to the number of packets received:

$$N_{rep} = \frac{N_f}{N_r}. \quad (1)$$

Like other security schemes, trust management schemes may also be vulnerable to many attacks such as bad mouthing attack. Hence, the trust management mechanism should consider these types of attacks also. In this work, in addition to attacks aimed at network, the attacks aimed at trust level are also considered. Trust value of a node is calculated based on direct trust information and indirect trust recommendations. The trust value of one node towards the other nodes in the network is updated periodically. Each node executes the trust protocol for finding the trust values of the other nodes by using the various trust metrics.

4. Enhanced multi-attribute trust (EMATP) protocol

The EMATP scheme performs trust management in three phases, namely trust evaluation at sensor node level, trust aggregation at SN level and trust update at CH level. The working principle of EMATP scheme is presented in figure 2. The first type of trust property based on connectivity is evaluated by observing the routing behaviour of a node. That is, how much capability a node has to deliver the data to the destination. We have taken MSR and elapsed time in a node (ETN) to measure this type of trust value. Here, MSR is the ratio of forwarded messages to received messages. Due to the data-centric nature of WSNs, modification attacks are very dangerous. Addressing this type of attacks is of greater importance. Elapsed time is measured as the elapsed time between the reception and the forwarding from a node. This property is mainly used to detect the address and the data modification attacks easily and quickly when compared with integrity checks, which take complicated procedures and longer time for checking. To the best of our knowledge, this is a novel work to defend against data or address modification attacks in an efficient manner.

The second type is evaluated by examining how a node behaves with its neighbours in the network. Here, we have taken CS and FS as two metrics. CS is measured by observing whether a node gives correct information about trust or not. FS is measured by observing whether the node

Figure 2. EMATP framework.

provides the same information or contradictory information to its different neighbours at the same time. This type of trust metrics defends against the attacks on trust management systems.

4.1 Trust evaluation at sensor node level

When node N_i and node N_j are one-hop neighbours, N_i evaluates the trust value of node N_j by direct calculation. This trust value $DT_{i \rightarrow j}$ is calculated by this formula as follows:

$$DT_{i \rightarrow j}(t1) = \sum_{\text{forall } P_i \in P} W_{P_i} * DT_{i \rightarrow j}^{P_i}(t1) \quad (2)$$

where P_i represents the various trust properties ($P = \text{MSR, ETN, CS, FS}$), W_{P_i} represents the associated weight of each property with the condition that sum of all these weights is equal to 1. When N_i and N_j are one-hop neighbours, $DT_{i \rightarrow j}(t1)$ is the trust value of N_j that N_i has at a particular time $t1$, and $DT_{i \rightarrow j}^{P_i}(t1)$ is the direct trust value of N_j that N_i has for a particular trust property P_i at time $t1$. When N_i and N_j are not one-hop neighbours, then there is no direct calculation by node N_i . In this case, node N_i uses the suggestions given by their neighbours where $k = 1, \dots, m$ and m is the number of nodes that are neighbours to N_i . The first two trust values can be calculated either by getting the information from a monitor, which is the CH, or by snooping or overhearing N_j . Next two properties $DT_{i \rightarrow j}^{\text{CS}}(t1)$ and $DT_{i \rightarrow j}^{\text{FS}}(t1)$ are estimated based on the information provided by N_j .

(a) $DT_{i \rightarrow j}^{\text{MSR}}(t1)$: This trust metric represents the trust value of node N_i towards node N_j based on the success of message forwarding event of N_j by direct observations. The MSR

value is calculated by monitoring the number of messages forwarded by N_j over the number of messages received by N_j using the following formula:

$$MSR = \frac{N_j^f}{N_j^r}. \quad (3)$$

Here, N_j^f and N_j^r represent, respectively, the number of messages forwarded and received by N_j . The calculated MSR value is compared to a threshold TH_{MSR} and finally the trust for this property is calculated using the equation

$$DT_{i \rightarrow j}^{\text{MSR}}(t1) = \begin{cases} DT_{i \rightarrow j}^{\text{MSR}}(t1 - \Delta t) & \text{if } MSR < TH_{MSR} \\ DT_{i \rightarrow j}^{\text{MSR}}(t1 - \Delta t) - \rho_{MSR} & \text{otherwise} \end{cases} \quad (4)$$

(b) $DT_{i \rightarrow j}^{\text{ETN}}(t1)$: This trust metric represents the trust value of node N_i towards node N_j based on the elapsed time between the reception and the forwarding of a message at node N_j . Node N_i calculates the ETN of N_j based on direct observations. The ETN value is calculated by monitoring the time at which a particular message is received by node N_j and how much time it takes to forward that message. This is the strongest evidence for data and address modification attacks. When compared with the previous techniques which use integrity check, this technique takes only little time and avoids complex calculations as found in integrity checks in the other techniques. Let the time T_j^r be the time at which the message is received by the node N_j , and T_j^f be the time when it is forwarded by the same node. Then, the ETN of N_j can be simply calculated as follows:

$$ETN_{i \rightarrow j} = T_j^f - T_j^r. \quad (5)$$

This value is compared to the acceptable threshold value and the trust value will be set accordingly using the equation

$$DT_{i \rightarrow j}^{\text{ETN}}(t1) = \begin{cases} DT_{i \rightarrow j}^{\text{ETN}}(t1 - \Delta t) & \text{if } ETN_{i \rightarrow j} < TH_{ETN} \\ DT_{i \rightarrow j}^{\text{ETN}}(t1 - \Delta t) - \rho_{ETN} & \text{otherwise} \end{cases} \quad (6)$$

Here the elapsed time is compared to a dynamic threshold. This threshold is based on the capability and resources of the node. Each node has a different threshold based on its capability. Hence, when a powerful node performs an attack, this attack can be addressed by the dynamic threshold of the node.

(c) $DT_{i \rightarrow j}^{\text{CS}}(t1)$: The trust value of node N_i towards node N_j is represented based on the assumption that a malicious node will give false recommendations. Hence, node N_i calculates the trust value of node N_j according to the information provided by N_j . Node N_i asks recommendations for some nodes that are already known to it. When the node N_j gives recommendations, N_i checks the consistency of

that recommendation with the calculated trust value. If more deviation is there, then N_i comes to the conclusion that N_j is malicious. Let $N = \{N_0, N_1, \dots, N_n\}$ be the set of nodes in a cluster, and ' n ' be the number of nodes in the cluster. $TN(i) = \{x_0, x_1, \dots, x_m\}$ is the set of trusted neighbours of N_i , and ' m ' is the number of trusted neighbours.

Consider the case where N_i calculates the trust of N_j . Node N_i has already known the trust values of its neighbours $TN(i)$, and now N_i wants to check whether N_j is trustworthy or not. Hence, N_i sends recommendation request to N_j and asks about $TN(i)$. N_j sends reply to N_i and this value is compared to actual trust values of $TN(i)$ calculated by N_i . Let $T_{j \rightarrow x_k}$ be the trust value of x_k given by N_j and $T_{i \rightarrow x_k}$ be the trust value of N_i towards x_k , where $k = 1, 2, \dots, m$ and TH_{CS} be the trust deviation threshold. If the calculated value is less than the threshold, then N_i considers N_j as a trusted one, otherwise the trust value is fixed as follows:

$$DT_{i \rightarrow j}^{CS}(t1) = \begin{cases} DT_{i \rightarrow j}^{CS}(t1 - \Delta t) & \text{if } \frac{\sum_{k=1}^m T_{j \rightarrow x_k} - \sum_{k=1}^m T_{i \rightarrow x_k}}{m} < TH_{CS} \\ DT_{i \rightarrow j}^{CS}(t1 - \Delta t) - \rho_{CS} & \text{otherwise} \end{cases} \quad (7)$$

(d) $DT_{i \rightarrow j}^{FS}(t1)$: This trust metric represents the trust value of node N_i towards N_j based on the assumption that malicious nodes normally provide contradictory information to their neighbours. To verify this, N_i first selects the trusted neighbours list based on the trust value in the last round. Neighbour nodes that have the highest trust value in the last round are considered as trusted neighbours. After selecting the trusted neighbours, N_i asks them to make a recommendation request to N_j for the node N_k that has the lowest trust value, i.e., this node should not be in the selected list. Based on the reply given to various nodes, N_i decides about N_j . If contradictory information is given by N_j to various nodes, then it is the malicious node.

Consider the case where N_i calculates the trust of N_j . N_i has already known the trust values of its trusted neighbours $TN(i)$ and untrusted neighbours $UN(i)$. Now, N_i wants to check whether N_j is trustworthy or not. N_i sends a request to its trusted neighbours $TN(i)$ to get recommendations for $UT(i)$ from N_j . All $TN(i)$ get recommendations for $UN(i)$ from N_j and send back to N_i . These recommended trust values given by N_j are compared for deviation, if any. Let $T_{j \rightarrow y_i, x_k}$ be the trust value of y_i given by N_j to x_k where $UT(i) = \{y_1, y_2, \dots, y_n\}$ be the set of untrusted neighbours of N_i . The consistency of the information is verified by finding the variance of $T_{j \rightarrow y_i, x_k}$ for all $i = 1, \dots, n$. If any one of the given recommendations heavily deviates with others, then N_i considers N_j as a malicious one and the trust value is fixed as follows:

$$DT_{i \rightarrow j}^{FS}(t1) = \begin{cases} DT_{i \rightarrow j}^{FS}(t1 - \Delta t) & \text{if } \forall i = 1..n, \text{Variance}(DT_{j \rightarrow x_k, y_i}) < TH_{fs} \\ DT_{i \rightarrow j}^{FS}(t1 - \Delta t) - \rho_{FS} & \text{otherwise} \end{cases} \quad (8)$$

4.2 Trust aggregation at sensor node level

Once the direct trust calculation has been completed, the SN estimates the final trust value by combining the individual trust values of each property. Each trust property is assigned a weight according to its significance. The individual trust values are added with their corresponding weight assigned according to the weight of each type of the trust value. The weight assigned to each trust value has an impact on the total trust value. Hence proper weight assignment is an important factor in trust calculation. The final trust value of node N_j is calculated by

$$T_{i \rightarrow j}(t1) = \alpha DT_{i \rightarrow j}^{MSR}(t1) + \beta DT_{i \rightarrow j}^{ETN}(t1) + \gamma DT_{i \rightarrow j}^{CS}(t1) + \delta DT_{i \rightarrow j}^{FS}(t1) \quad (9)$$

where $DT_{i \rightarrow j}^P(t1)$ is the direct trust value of node N_i towards node N_j , and α, β, γ and δ are the weight values of individual trust properties of direct trust with the condition that sum of α, β, γ and δ is equal to 1. Whenever a node calculates the trust value, it finds the deviation between the previous trust value and the current trust value. If this deviation is greater than a minimum level, then the node decides to calculate the trust for the next round. Hash-based security is meant for the integrity of data. Here the data modification attack is addressed by elapsed time as a metric. Hence, when compared with hash-based security, this takes smaller time and has lesser computational complexity.

Even though the parameters are inconsistent, here the parameters are compared to an acceptable threshold value. Hence it is accepted only to a certain level and when it crosses the limit, this may be considered as misbehaviour. Moreover, the trust value does not depend on a single parameter; it depends on multiple parameters, and hence other parameters can manage these inconsistencies.

4.3 Trust update at cluster head level

Each SN sends the calculated direct trust value of its neighbours to the CH. After receiving this information, the CH combines the trust values given by different nodes, updates the old trust value and finds the final trust value of each node. Based on this, the CH finds the malicious nodes that have a trust value below a minimum threshold TH^t . Then, it sends the list of malicious nodes to BS, and now BS rejects the nodes that are in the list. Let $T_{i \rightarrow j}(t1)$ be the trust value of N_j at time $t1$, $DT_{i \rightarrow j}^P(t1)$ be the direct trust value calculated by any node N_i towards the other node N_j . The new trust value at time $t1$ is updated by considering the previous trust value at time $t1 - \Delta t$ and it is evaluated by the following formula:

$$T_{i \rightarrow j}(t1) = (1 - \rho)e^{-\omega \Delta t} DT_{i \rightarrow j}(t1 - \Delta t) + \rho DT_{i \rightarrow j}(t1). \quad (10)$$

Here, $e^{-\omega \Delta t}$ is the exponential time decay factor, and ρ is the weight of trust value. Whenever the new trust value is updated, a new list is prepared and sent to the BS. Trust of a node is calculated based on its behaviour. If the node is in sleep mode, there is no information available on its behaviour. Hence Δt depends on the active time of a node. When Δt increases, the previous trust will have lower importance and the trust value is decreased by multiplying with a time decay factor. When the node goes to the sleep mode, its previous trust value is stored and after it moves to the active mode, the new trust value will be calculated based on its behaviour. Here periodical reporting is considered.

Algorithm for direct trust calculation

1. For all $N_i \in N$, do the following
2. For each $m \in NH(i)$ do the following
3. Find the number of packets received and forwarded by 'm'
4. Calculate the trust value for MSR property using Eq. (4)
5. Determine the elapsed time between receiving a packet and forwarding the same by 'm' using Eq. (5)
6. Calculate the trust value for ETN property by Eq. (6)
7. N_i asks trust recommendations R_k from 'm' for all its trusted neighbours $k \in TN(i)$
8. Find the deviation of R_k from $T_{i \rightarrow k}$, compare it to minimum threshold TH_{cs} and set the trust value for CS property using Eq. (7)
9. N_i asks $k \in TN(i)$ to get recommendations from 'm'
10. Find the difference in the recommendations, compare it to the minimum threshold TH_{fs} and set trust value for FS property using Eq. (8)
11. Aggregate the various trust property values to get the final trust value by Eq. (9)

5. Results and discussion

Experiments were conducted in Network Simulator-2(NS-2) [21]. Results of the simulation are given to demonstrate trust determination accuracy and the performance of the proposed trust protocol. In this simulation, the efficiency of the proposed protocol is evaluated in an insecure environment. The performance of the proposed work is compared to that of the hierarchical trust management protocol-based routing (HTBR) [15] and traditional geographical routing (GR). Detection accuracy is compared to those of the existing anomaly-based techniques in weighted summation (WSIDS) [18] and hierarchical trust management (HTIDS) [15].

5.1 Environment set-up

Nodes are deployed randomly in an area. Each transmission considers a pair of source and destination nodes with CBR traffic. We consider three scenarios. In the first one, routing is done without considering the trust of nodes. Data pass through the malicious nodes also. In the second scenario, the trust of each node is considered based on hierarchical trust management. The third one is the proposed protocol in which the trust of each node is calculated by considering multi-attributes based on routing and social behaviour of each node. We assume that the number of malicious nodes in the network is only an acceptable fraction of the total number of nodes in the network. Simulation is performed with different number of nodes and different number of attackers. Four performance metrics are taken for simulation.

(a) *Delivery ratio*, which is a measure of the number of packets received by the destination node among the total number of packets generated by the source node.

(b) *Average delay*, which is a measure of end-to-end delay between source and destination nodes.

Figure 3. (a) Impact of malicious nodes on deliver ratio, (b) impact of malicious nodes on average delay and (c) impact of malicious nodes on message overhead.

[illegible]

Table 3. Memory requirement at CH in bytes.

Node ID	Trust value	ρ	ω
2	4	4	4

Figure 5. Number of clusters versus memory requirement.

bytes. The storage requirement of each node is given in the following tables 2 and 3.

In figure 5, storage requirements for different number of clusters are plotted. When the number of clusters is increased, memory requirement is reduced at each node, since the number of neighbours is reduced.

5.5 Analysis of computational complexity

When data fusion occurs at an intermediate node, the data from the trusted nodes are considered among the total number of nodes in the network. Even though the number of nodes gets increased, all the nodes are not taken into account for data fusion. Moreover data fusion is in the form of simple calculations, like minimum, maximum or average. Thus, this calculation does not increase the computational complexity.

6. Conclusion

A trust-based IDS that uses multi-attribute trust metrics to improve the intrusion detection accuracy in WSN is proposed in this paper. In enhanced multi-attribute trust protocol (EMATP), trust values are calculated by using QoS and effective social parameters, and the trust values are compared to those of the hierarchical trust management protocol based on weighted sum approach. This is

compared for delivery ratio, average delay and communication overhead. Results show that the proposed protocol has higher delivery ratio than that of trust-based geographic routing and smaller average delay and overhead. When applied to intrusion detection, it is proved that this method provides more security in terms of detection accuracy and false alarm rate.

References

- [1] Butun I, Morgera S D and Sankar R 2013 A survey of intrusion detection systems in wireless sensor network. *IEEE Commun. Surv. Tutor.* 16(1): 266–282
- [2] Deng X, Wu R, Wang W and Bu R 2013 An intrusion detection system for cluster based wireless sensor networks. *Inf. Technol. J.* 12(9): 1764–1771
- [3] Theodorakopoulos G and Baras J S 2006 On trust models and trust evaluation metrics for ad hoc networks. *IEEE J. Sel. Areas Commun.* 24(2): 318–332
- [4] Cayirci E and Rong C 2009 *Security in wireless adhoc and sensor networks*. New York: Wiley
- [5] Padmavathi G and Shanmugapriya D 2009 A survey of attacks, security mechanisms, and challenges in wireless sensor networks. *Int. J. Comput. Sci. Inf. Technol.* 4(1–2): 117–125
- [6] Fuchsberger A 2005 Intrusion detection systems and intrusion prevention systems. *Inf. Secur. Tech. Rep.* 10(3): 134–139
- [7] Wei Z, Tang H, Yu F R, Wang M and Mason P 2014 Security enhancements for mobile adhoc networks with trust management using uncertain reasoning. *IEEE Trans. Veh. Technol.* 63(9): 4647–4658
- [8] Qassim Q, Patel A and Mahd-Zin A 2014 Strategy to reduce false alarms in intrusion detection and prevention systems. *Int. Arab J. Inf. Technol.* 11(5): 500–506
- [9] Boukerch A, Xu L and El-Khatib K 2007 Trust-based security for wireless ad-hoc and sensor networks. *Comput. Commun.* 30(11–12): 2413–2427
- [10] Yao Z, Kim D and Doh Y 2006 PLUS: parameterized and localized trust management scheme for sensor networks security. *Proceedings of the IEEE International Conference on Mobile Ad hoc Sensor Systems*, pp. 437–446
- [11] Yu Y, Li K, Zhou W and Li P 2012 Trust mechanisms in wireless sensor networks: attack analysis and countermeasures. *J. Netw. Comput. Appl.* 35(3): 867–880
- [12] Ganeriwal S and Srivastava M B 2004 Reputation-based framework for high integrity sensor networks. *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Washington, USA, pp. 66–77
- [13] Shaikh R A, Jameel H, Auriol J B and Lee H 2009 Group-based trust management scheme for clustered wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* 20(11): 1698–1712
- [14] Xiang G, Jianlin Q and Jin W 2012 Research on trust model of sensor nodes in WSNs. *International Workshop on Information and Electronics Engineering. Proc. Eng.* 29: 909–913

- [15] Bao F, Chen I R, Chang M and Cho J H 2012 Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. *IEEE Trans. Netw. Serv. Manag.* 9(2): 169–183
- [16] Ram Prabha V and Latha P 2014 Game theory model for intrusion detection in WSN using hierarchical trust management. *Int. Rev. Comput. Softw.* 9(6): 1055–1064
- [17] Kim T K and Seo H S 2008 A trust model using fuzzy logic in wireless sensor network. *Int. J. Electr. Comput. Energetic Electron. Commun. Eng.* 2(6): 1051–1054
- [18] Hu H and Chen Y 2009 Weighted trust calculation based malicious node detection for wireless sensor networks. *Int. J. Inf. Comput. Secur.* 3(2): 132–149
- [19] Liu K, Abu-Ghazaleh N and Kang K D 2007 Location verification and trust management for resilient geographic routing. *J. Parallel Distrib. Comput.* 67(2): 215–228
- [20] Loo C E, Ng M Y, Leckie C and Palaniswami M 2006 Intrusion detection for routing attacks in sensor networks. *Int. J. Distrib. Sens. Netw.* 2(4): 313–332
- [21] Network Simulator: <http://www.isi.edu/nsnam>