Wireless Personal Communications (2018) 103:2501–2513
https://doi.org/10.1007/s11277-018-5942-8

A Distributed Trust Management Mechanism for the Internet of Things Using a Multi-Service Approach

Carolina Veronica Lezama Mendoza[1] · **João Henrique Kleinschmidt**[1]

Published online: 12 September 2018

Abstract
In the Internet of Things (IoT) heterogeneous devices can cooperate and communicate to provide or require determined services. A multi-service IoT is vulnerable to many types of malicious attacks. A trust management scheme is a strategy to establish trust between devices. In this work we propose a distributed trust management model for IoT using direct and indirect observations. The trust of a node is computed by the service quality and recommendations from neighbors. The nodes locally compute the trust of their neighbors, without the need of a central entity. We implemented the proposed strategy in the Cooja simulator of the Contiki operating system to analyze the performance of the trust model. We performed simulations using different number of malicious nodes performing the bad mouthing attack. Simulation results show the trust management scheme is able to detect malicious nodes in the network.

Keywords Internet of Things · Trust management · Security · Multi-service

1 Introduction

The Internet of Things (IoT) is a network formed by heterogeneous devices with different processing capabilities which can cooperate and communicate in an intelligent environment [1, 2]. For the next years, we expect some billions of devices connected to the IoT. It is a global network where smart objects, such as computers, smartphones, sensors, actuators and other everyday life devices communicate and provide information in real time. IoT is one of the enabling technologies for smart cities and has been developed in applications in the areas of agriculture, health, intelligent transportation, smart grids, industrial monitoring, among others. This integration of many everyday life devices from heterogeneous network environments brings a great challenge to information security [3, 4]. The devices often use wireless communications, hence more vulnerable to malicious attacks.

✉ João Henrique Kleinschmidt
joao.kleinschmidt@ufabc.edu.br

Carolina Veronica Lezama Mendoza
carolina.mendoza@ufabc.edu.br

[1] Universidade Federal do ABC, Santo André, Brazil

Trust management (TM) is a mechanism to establish trust between two individual nodes and it can provide solutions to problems such as key management, authentication mechanisms and secure routing [5]. TM systems can be applied as a factor to promote trust in the communication between unknown entities, as in the case of IoT. These systems constitute a way to encourage honest collaboration between nodes (smart objects), while reducing the importance of participation by malicious nodes or anomalous functioning and may eventually serve as a basis to isolate them completely of the network. TM has been widely studied in many network environments such as peer-to-peer networks, grid, ad hoc and wireless sensor networks [5–7]. In current trust management mechanisms for IoT we do not have all requirements needed for a functional implementation in this context. There are only some works with focus in the IoT [8, 9].

New trust management mechanisms have to be developed, since many objects in the IoT have limited resources in processing, storage and power, the network is heterogeneous and different services have to be offered. An efficient lightweight trust management mechanism that fits the characteristics of smart objects and the multi-service IoT is indispensable [8–11].

In [12] the authors proposed a model of a centralized trust management scheme for a multi-service environment, where a centralized trusted entity evaluates the trust of each node in the network. In [13] we proposed a distributed TM scheme where the decision of the trust value is computed locally by the nodes of the network, without the need of a central device. The trust value is based on direct observations, depending on the assistance of services by the collaborative nodes. The TM model was evaluated under On–Off and selective attacks [13, 14]. In this work we extend the model presented in [13] and [14] using indirect observations in the computation of trust. The novelty of the proposed TM mechanism is to compute the trust of a node locally using direct and indirect information. The indirect information is based on the recommendation of nodes about their neighbors. We study the effectiveness of the proposed system in the presence of bad mouthing attacks. In these attacks, malicious nodes send incorrect recommendations about their neighbors, destroying the reputation of good nodes or increasing the trust of malicious nodes.

The paper is structured as follows: in Sect. 2 we discuss trust management concepts for IoT and related work; the proposed model with direct and indirect observations is presented in Sect. 3; Sect. 4 shows the simulation results and finally, Sect. 5 gives the final considerations.

2 Trust Management for the Multi-service Internet of Things

Trust may be described as a two-way relationship between two parties, called trustor (the one which trusts) and trustee (the one that is trusted) in a given scope or context [5]. It is assumed that the trustor entity has the ability to make assessments and make decisions based on information received or past experiences. The trustee can be a person, organization, physical entity, or even abstract notions such as data and information. The main purpose of TM systems is to reduce the opportunism and vulnerability between unknown partners, so that they can take risks, build trust and decide if they want to interact. Trust is a time sensitive concept, which can evolve positively or negatively according to the behavior of the entities. TM are the mechanisms to evaluate, establish, maintain and revoke the trust between devices of the same or different networks within the IoT environment [6, 10, 11]. In the literature, many trust

management systems [5–7] have been proposed for wireless networks, such as ad hoc and sensor networks. More recently, some work in trust management has been studied in the context of IoT [8–17]. In [10], the author discusses some questions about trust in an IoT environment from a human perspective. The paper discusses if humans can trust devices in IoT, not if devices can trust other devices. In [8] and [9] the authors present a survey on TM schemes in IoT and present some research challenges in this area. While [8] presents a general survey, Guo et al. [9] focus on TM models for a multi-service approach, based on five dimensions: trust composition, trust propagation, trust aggregation, trust update and trust formation.

Trust composition is the components to consider in the trust evaluation: quality of service (QoS) and social trust [9]. Trust propagation is how the trust is propagated to the nodes in the network; it can be distributed or centralized. Trust aggregation is how to collect evidence for the evaluation of trust: direct (self-observations) or indirect (observations from other nodes). The main trust aggregation techniques include weighted sum, belief theory, Bayesian inference, fuzzy logic and regression analysis. Trust update is about when trust is updated: event driven (trust is updated after an event) and time driven (trust is updated by periodical collection of evidences). Finally, trust formation is how the value of trust if formed: single trust or with multi-trust properties. Single trust uses only one property, such as service quality. Multi-trust considers that trust is multi-dimensional and the value of trust must be formed by multiple properties.

Lacuesta et al. [10] proposed a TM scheme based on the concept of communities formed by nodes in an ad hoc network, where trust is formed based on their initial communities or companies. In Bao et al. [15] and Chen et al. [16] different metrics for a TM system are proposed, including the cooperation to provide a service and recommendations. The TM model uses service quality and social similarity to rate a recommender. It is distributed and uses Bayesian inference to aggregate direct observations and social similarity weighted sum to aggregate recommendations. The authors in [17] proposed a TM model with a simple game approach which achieves Bayes equilibrium. Chen et al. [18] proposed a distributed TM scheme for IoT based on fuzzy reputation using nodes recommendations. Nitti et al. [19] proposed a TM model using hybrid characteristics for trust propagation (centralized and distributed), considering also QoS and social trust. It is a multi-trust scheme combining centrality trust (social trust property) with service quality trust (using direct and indirect observations).

The work by Saied et al. [12] considers a context-aware and multi-service approach for IoT, where trust is evaluated based on self-observations and observations from other nodes (recommendations). In a multi-service IoT, nodes can provide different services. Each service has a different cost in terms of resources consumption of the nodes. The trust in [12] is not computed at individual nodes, but in a centralized trust manager. The scheme is evaluated in the presence of bad mouthing, selective repeat, and On–Off attacks. The option for a centralized scheme implies a trust management server responsible for trust computational load. However, it is not possible to implement such servers in all scenarios of IoT. The scheme proposed in [13, 14] is similar to [12], but using a distributed approach instead of a centralized strategy, eliminating the need of a central server. The scheme uses only direct observations and it was analyzed in the presence of On–Off and selective attacks. Our work extends the TM model of [13, 14] by considering also indirect observations and evaluating the model under bad mouthing attacks.

3 Proposed Distributed Trust Management Scheme for a Multi-Service IoT

In this Section we present the proposed trust management model based in [13]. It uses direct observations, as described in [13], but also indirect observations. Thus, this section reviews the TM model of [13] and describes the additional phases of the model. Using the classification of [9], our model is characterized by using QoS for trust composition, distributed for trust propagation, weighted sum with direct and indirect observations for trust aggregation, event driven for trust update and single trust for trust aggregation. The decision to trust or not in some device in the IoT environment has to consider many factors and characteristics. In some cases, there are different levels of trust between entities. One device may be considered a trusted entity for a particular task, but not for others. In a IoT scenario with heterogeneous devices providing different types of services, not all of them have the same resource capabilities. Thus, some devices may not provide more resource consuming services.

The TM model was developed with the goal that the nodes are able to manage the trust values with respect to the services provided by the other nodes. We implemented a distributed mechanism where all devices have the same features. The TM model is able to identify whether a node has a malicious behavior or not in order to eliminate this node of the network. The proposed scheme does not have a central entity to manage the communications or trust values among nodes. Therefore, each node has an autonomous and independent behavior in the trust evaluation. Each node in the network is able to provide a different number of services.

The trust management model is divided in 2 sections and 6 phases, as shown in Fig. 1: using direct information: neighbor discovery, service request and trust computation; using indirect information: send of a trust table, evaluation of the recommendation and trust value updates.

3.1 Phase 1: Initial Communication for Neighbor Discovery

At the initial setup of the network, all the nodes assign a trust value = 0 (zero) for all its neighbors. This initial value means ignorance of the node´s behavior or unknown trust. Initially all nodes are unknown nodes and the initial trust value is filled with 0 (zero) in the

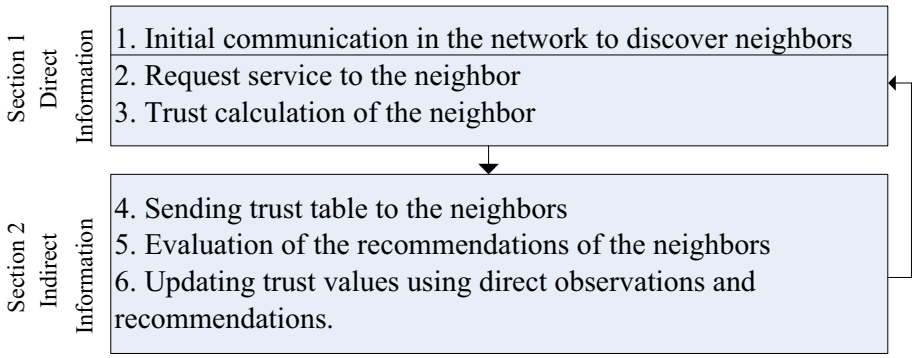


Fig. 1 Phases of the distributed trust management model

trust table of the node. The network performs the neighbor discovery process. The nodes announce their presence by periodically sending announcement packets by broadcasting it to its neighbors. These packets are used to populate the neighbor table with neighbors. The trust table will be filled with information of the nodes that are in the same transmission range giving to the neighbors initial trust values = 0 (zero).

3.2 Phase 2: Request Service to the Neighbor

Each service has a reward and punishment value each time it is provided or not to the neighbor which requested it. When a node provides with success a requested service, it receives a reward and it receives a score based on Eq. (1):

$$N_j = 1 * W_s, \tag{1}$$

where N_j is the score to a node for providing or not a requested service j and W_s is a weight assigned for each service calculated by Eq. (2):

$$W_s = S_j * \sigma \tag{2}$$

where S_j is a value assigned to a specific service and σ is an adjust factor which vary in $0 < \sigma < 1$. This adjust factor is the expected rapidity of change in the node. The higher the value assigned to σ, faster the trust will converge, while a lower value assigned to σ the trust will converge slowly. The services are valued according to their resource consuming capabilities; each service has different requirements of energy, memory and processing in a node. The services that require more processing capacity have a high value of S_j and the services which do not require many resources have a low value of S_j.

In the same way, when a node does not provide with success a requested service, it receives a punishment and its score is based on Eq. (3):

$$N_j = -2 * W_s \tag{3}$$

The node is punished twice the value of the weight for each service. The TM model punish any undesired behavior (malicious or not).

3.3 Phase 3: Trust Calculation of the Neighbor

The trust value T_{np} of the node n computed by node p is the sum of the scores of all requested services:

$$T_{np} = \sum N_j, \quad \text{where } -1 < T_{np} < 1 \tag{4}$$

Trust is a value ranging from 1 to − 1, where 1 is equivalent to the maximum score of trust and − 1 is the minimum score of trust that a neighbor node can reach in the trust table. When a node A request a service to node B and the node B provide properly that service to the node A, the node B is rewarded by node A. The reward is an increase in the trust value in the trust table of the node A. The increase is dependent of the service provided. The trust value of that node changes according to Eqs. (1–4). When nodes do not provide the services, they are punished by the node requesting the service. The trust value of the node decreases each time that the node does not provide successfully a requested service.

So, in the TM scheme, negative values mean a malicious behavior. A trust value of -1 (or close to it) means a high distrust in the node to provide a service. To be considered honest, a node must have a trust value of 1 (or close to it).

3.4 Phase 4: Sending Trust Table to the Neighbors

A node sends the trust table to all its neighbors during an interval of t seconds. There is a trade-off between high and low values of t. A slow value of t could lead to many transmissions and causing an overhead in the network, but it maintains the nodes updated with the behavior of the network. A high value of t does not capture instantly the behavior of a neighbor node, but the overhead introduced in the network is lower.

We implemented the use of some adjustment factors that will allow us to evaluate the values received from the neighbors with respect to the current trust values that the node has observed from previous communications. We defined $Trust^{node}_{neighbor}$ as the trust value of a *neighbor* stored in the trust table of *node*. This value is obtained from direct communication between them. The β threshold is used to evaluate whether the trust value of the *neighbor* who is sending the trust table is good enough to take into account in the assessment process of the recommendations; otherwise these values will be discarded (Eq. 5). A high value of β means we trust only in the recommendations of nodes with a high value of trust. With a low value of β the recommendations of nodes with low trust values will not be considered. The node that receivers a trust table from its neighbor only will pass to next phase if the following condition is satisfied:

$$Trust^{node}_{neighbor} \geq \beta \tag{5}$$

3.5 Phase 5: Evaluation of Recommendations of the Neighbors

In the assessment process of the recommendations we used Qr as the quality of the recommendation of the node. Qr is a value that allows to identify the consistency of the recommendations (trust values) received. A trust value very different from the current value means inconsistency in the recommendations and therefore could be an attack. Qr is calculated from the current trust value of the node and the recommendation received from the neighbor. The value obtained in Qr will be used in the process of updating the trust of the recommended node and the trust of the node who sent the recommendation. If $Qr = 1$ means that the values of trust are identical (report is coherent). Otherwise if $Qr = -1$ the values of trust are opposite (Qr is contradictory). We define Qr as:

$$Qr = -|Trust^{neighbor}_{node\_X} - Trust^{node}_{node\_X}| + 1 \tag{6}$$

where $Trust^{neighbor}_{node\_X}$ is the recommended trust value of the *node_X* by the *neighbor* that is sending the trust table and $Trust^{node}_{node\_X}$ is the current trust value of *node_X* stored in the trust table of *node* which was obtained from direct communication between *node* and *node_X* or previous recommendations.

The α threshold is used to analyze the result obtained in Qr with Eq. (6). If the value obtained in Qr is higher than α, then the recommended trust value pass to the next phase of the evaluation process (Eq. 7) with Qr as a good recommendation. If Qr is lower than α, the recommendations are bad.

$$Qr \geq \alpha \tag{7}$$

3.6 Phase 6: Updating Trust Values Using Direct Information and Recommendations

The phase of updating trust values use two types of information: the information obtained from direct communication between nodes and the trust tables from other nodes. We pass to the phase 6 once that the verification of Qr in (7) is performed. This evaluation has two types of assigning values:

- Reward process as shown in Eqs. (8) and (9)
- Punish process shown in (9)

We update the trust values of the recommended node if and only if the result in (7) was true. *NewTrust* is defined as the new trust value that we will be assigned to the *node_X,* which is the recommended node by the neighbor. We defined the update process as shown in Eq. (8):

$$NewTrust^{node}_{node\_X} = Trust^{neighbor}_{node\_X} + \left(Trust^{node}_{node\_X} * Qr\right) \tag{8}$$

Since the recommendation received from the neighbor was a good recommendation for the process of updating trust values of the node, we reward the node that sent the trust table. In our model we increment the trust value of the neighbor who sent the trust table using the value obtained in the calculation process of Qr in (6) and the adjust factor θ, as shown in Eq. (9). While higher the value assigned to θ, the trust value will increase fast and while lower the value assigned to θ, the trust value will increase slowly.

$$NewTrust^{node}_{neighbor} = Trust^{node}_{neighbor} + (Qr * \theta) \tag{9}$$

Moreover if the result of (7) does not pass the verification of the expected value, then the neighbor is punished. This process is carried out also by Eq. (9). In case of the punishment to the node we expect that the value of Qr be negative and therefore the Eq. (9) will subtract the result of the multiplication of Qr and θ. The trust value of the neighbor is decreased in the trust table of the node using the same value obtained in quality recommendation process in (6). All these calculations are executed by each node when they receive the updates from the neighbors in the same transmission range.

4 Simulation Results

In this section we show the results obtained for the proposed trust management model. We implemented the trust management scheme in the COOJA simulator included in Contiki Operating System [20]. In the simulations is used the Unit Disk Graph Medium (UDGM) as radio model, the ContikiMAC as radio duty cycle (RDC) mechanism and CSMA/CA (Carrier Sensor Multiple Access with Collision Avoidance). The simulated network is formed by 50 Tmote Sky nodes randomly distributed. Each node requests a service to a random neighbor with a random interval between 0 and 60 s. The nodes are capable of providing 3 types of services with the following values: $S_1 = 0.1$, $S_2 = 0.05$ and $S_3 = 0.02$. The value of the adjust factors and simulation parameters are: $\sigma = 0.1$, $\alpha = 0.2$, $\beta = 0.4$ and $\theta = 0.5$. The simulation time is 2 h and each simulation is repeated 15 times. The results

presented are the average for the 15 simulations. The trust table with the recommendations is sent once per minute. In a network where nodes send recommendation to others, there is a risk of receiving wrong recommendations, known as a bad mouthing attack. In this type of attack the malicious nodes are able to destroy the reputation of well behaved nodes by sending reports with false recommendations to the others nodes in the network. Also, a malicious node may boost the trust values of malicious peers, by sending false recommendations with high values of trust. We implemented this attack in 3 simulation scenarios with 10%, 20% and 30% of malicious nodes in the network. In our simulation, a malicious node provides dishonest recommendation of honest nodes (sending low trust values for good nodes).

Figure 2 shows the network topology, being 10% malicious nodes performing the bad mouthing attack. For the analysis of the results of this scenario, we selected the node 32 to verify the behavior of the nodes. The node 32 has a total of 30 neighbors in its range. Three of these neighbors are malicious nodes (nodes 3, 4 and 5). Although we present only the analysis of node 32, the other nodes have a similar behavior in the identification of malicious and honest nodes.

Figure 3 shows the behavior of the trust values for the malicious nodes. The node 32 is able to identify the malicious behavior by using the direct and indirect information. The initial trust values of the malicious nodes are positives values; this does not represent a failure to recognize the malicious behavior but is part of the process of identifying a malicious node. In the bad mouthing attack the malicious nodes provide appropriately the services requested by other nodes. So, if node 32 requests a service to a malicious node, it will receive a positive answer. Only after receiving trust tables from its neighbors is possible to identify the malicious behavior. The maximum positive score that the malicious nodes reached in this scenario is 0.3, which is a low score of trust and, therefore, they cannot be considered a trusted node.

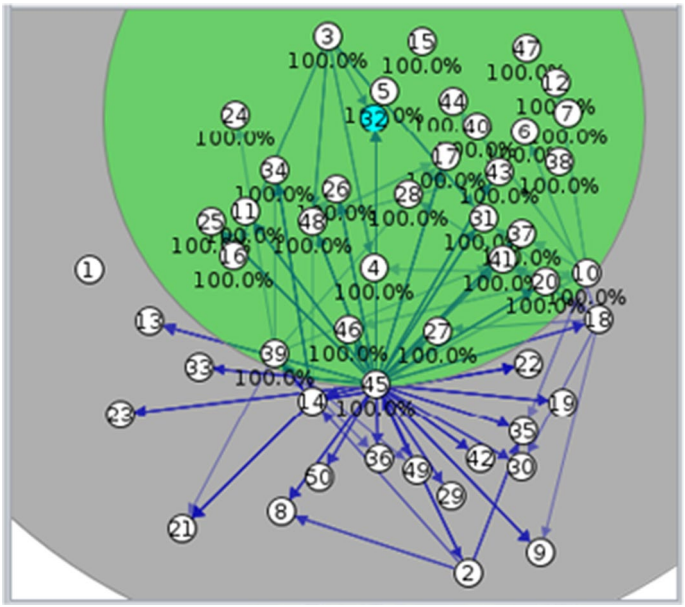


Fig. 2 Scenario 1—network topology with 10% of malicious nodes

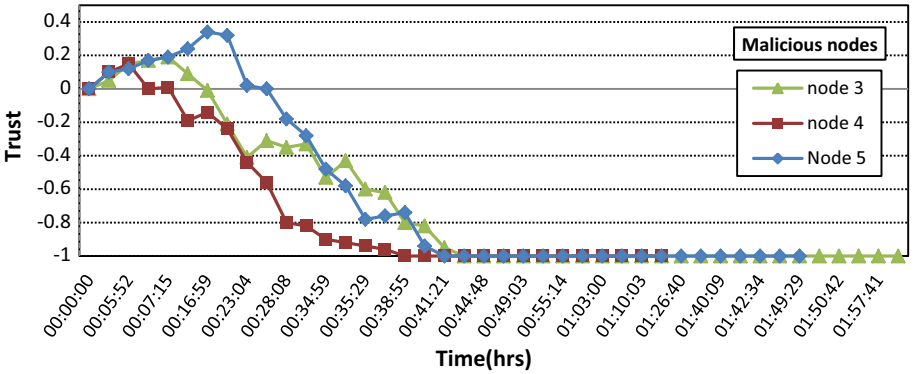


Fig. 3 Trust values for malicious neighbors (scenario 1)

The node takes a total average time of 40.33 min to assign the minimum score of trust to the malicious nodes and 90 min to assign the maximum score of trust to the well-behaved nodes. In Fig. 4 is shown the average time where the node 20 is a good node and the nodes 3, 4 and 5 are malicious nodes. In a scenario with a more frequent update of the trust table the time to detect a malicious node will decrease. The main motivation is to analyze whether the trust management scheme detects the malicious activity. The higher the interval update of the trust table, the lower the time to detect a malicious node. However, a more frequent update leads to high traffic in the network and it consumes more resources of the nodes. It is important to note that for a node to be considered malicious, it is not necessary to reach the value -1 (for instance, with trust equal to -0.7 the node may be considered malicious).

Figure 5 shows the network topology used in a simulation with 20% of malicious nodes. The node 32 is the selected node for the analysis of the results. It has a total of 29 neighbors in its range; 5 of these neighbors are malicious nodes (nodes 1,5, 6, 7 and 9). Figure 6 shows the time to assign the maximum value of distrust to the malicious nodes. The node 32 calculate the trust values depending on the service that is provided by malicious node; once node 32 begins to receive recommendations values, the trust values with respect to the malicious node begins to decrease gradually until reaching the maximum distrust (-1). In the same way, the honest nodes will reach the maximum value of trust (1). The average scores obtained by the node 32 for some neighbors are presented in Fig. 7. As can be observed the malicious nodes are detected in an average time of 48 min, while the honest nodes are detected with maximum value in 78 min.

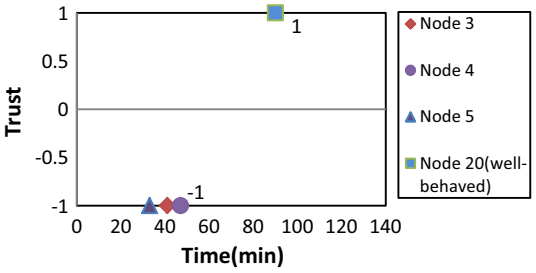


Fig. 4 Average time to classify the node behavior (scenario 1)

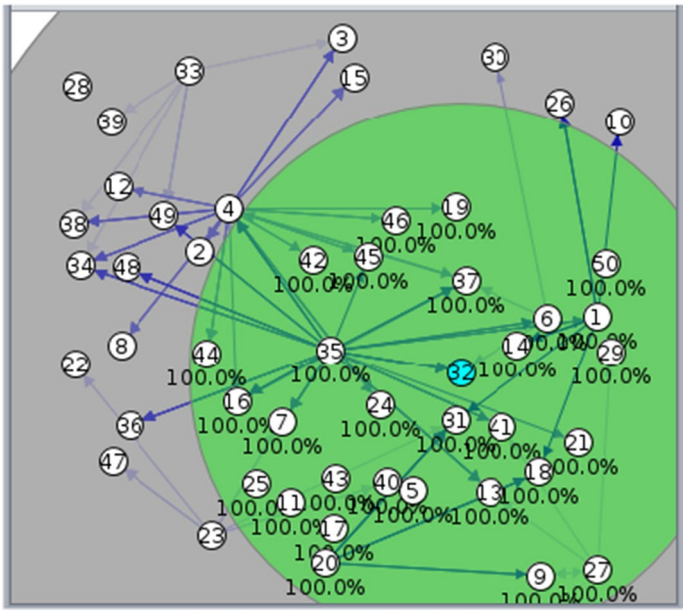


Fig. 5 Scenario 2—network topology with 20% of malicious nodes

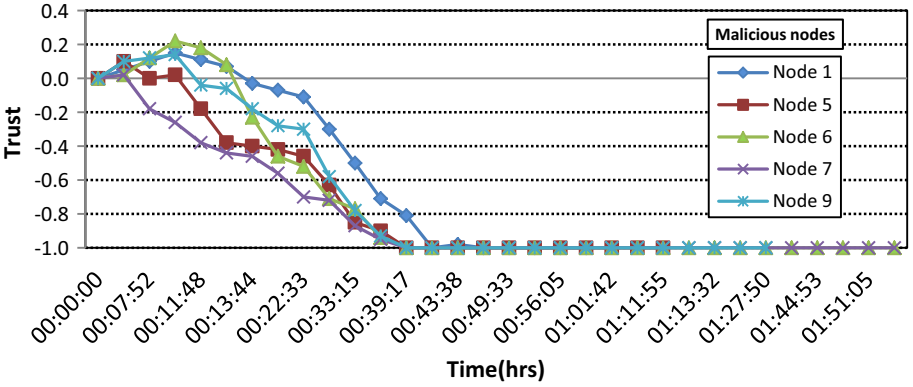
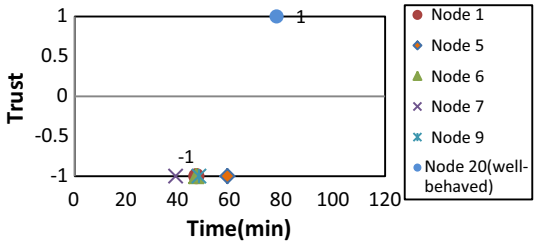


Fig. 6 Trust values for malicious neighbors (scenario 2)

Fig. 7 Average time to classify the node behavior (scenario 2)



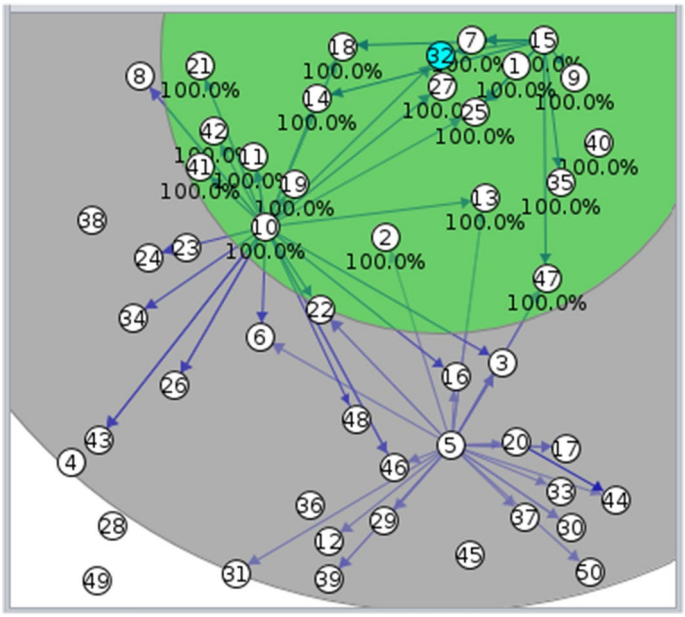


Fig. 8 Scenario 2—network topology with 30% of malicious nodes

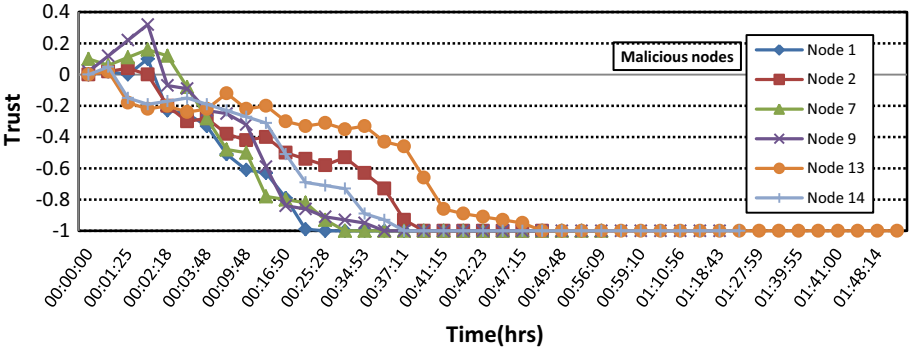


Fig. 9 Trust values for malicious neighbors (scenario 3)

Figure 8 shows the network topology for a scenario with 30% of malicious nodes. Again the node 32 is the selected node for the analysis of the results. The node 32 has a total of 19 neighbors; 8 of these neighbors are malicious nodes in the same transmission range (nodes 1, 2, 7, 9, 10, 13, 14 and 15). Figure 9 shows the time to assign the minimum value of trust to the malicious nodes. Even when the number of malicious node is increased to 30%, the trust model still has a good performance to detect the malicious behavior in the network. Node 32 takes a total average time of about 52.33 min to assign the maximum score of distrust to the misbehavior nodes and 1.30 h to assign the maximum trust to the well-behaved nodes (Fig. 10). In [12] and [13] were used only direct observations (phases 1 to 3 of the

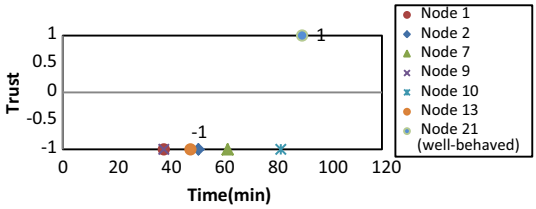


Fig. 10 Average time to classify the node behavior (scenario 3)

TM model) to detect a malicious behavior (On–Off and selective attacks). So our proposed model with direct observations and recommendations may detect malicious nodes performing other types attacks, not only the bad mouthing attack.

5 Conclusions

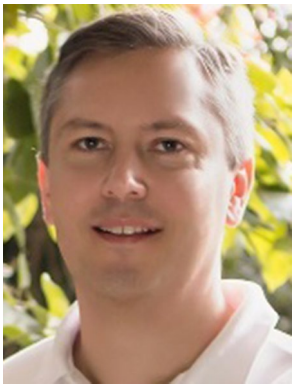
In this paper we proposed a distributed trust management model for multi-service IoT using direct and indirect observations. The TM scheme assigns positive scores for honest nodes and negative scores for malicious nodes, using direct interactions between nodes (services requests) and recommendation from neighbors (by exchanging trust tables). We implemented malicious nodes performing the bad mouthing attack to analyze the effectiveness of the model. The obtained results show the proposed TM model detects malicious behavior in the network, considering topologies with 10% until 30% of malicious nodes. This model may be used to detect other common attacks in the IoT, such as On–Off and selective attacks. Future works may improve the TM model and evaluate its performance in the presence of other types of attacks.

References

1. Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks, 10*(7), 1497–1516.
2. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks, 54*(15), 2787–2805.
3. Roman, R., Najera, P., & Lopez, J. (2011). Securing the Internet of Things. *IEEE Computer, 44*(9), 51–58.
4. Roman, R., Zhou, J., & Lopes, J. (2013). On the features and challenges of security and privacy in distributed Internet of Things. *Computer Networks, 57,* 2266–2279.
5. Cho, J. H., Swami, A., & Chen, R. (2011). A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials, 13*(4), 562–583.
6. Chang, K. D., & Chen, J. L. (2012). A survey on trust management in WSNs, Internet of Things and future internet. *KSII Transactions on Internet and Information Systems, 6*(1), 5–23.
7. Han, G., Jiang, J., Shu, L., Niu, J., & Chao, H. C. (2014). Management and applications of trust in wireless sensor networks: A survey. *Journal of Computer and System Sciences, 80*(3), 602–617.
8. Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. *Journal of Network and Computer Applications, 42,* 120–134.
9. Guo, J., Chen, I., & Tsai, J. P. (2017). A survey of trust computation models for service management in Internet of Things systems. *Computer Communications, 97*(1), 1–14.
10. Køien, G. M. (2011). Reflections on trust in devices: An informal survey of human trust in an Internet-of-Things context. *Wireless Personal Communications, 61*(3), 495–510.
11. Lacuesta, R., Palacios-Navarro, G., Cetina, C., Peñalver, L., & Lloret, J. Internet of things: Where to be is to trust. *EURASIP Journal on Wireless Communications and Networking*, 1–16, 2012.
12. Saied, Y. B., Olivereau, A., Zeghlache, D., & Laurent, M. (2013). Trust management system design for the Internet of Things: A context-aware and multi-service approach. *Computers & Security, 39,* 351–365.

13. Mendoza, C. V. L., & Kleinschmidt, J. H. (2015) Mitigating On–Off attacks in the internet of things using a distributed trust management scheme, *International Journal of Distributed Sensor* Networks, vol. 2015, article ID 859731.
14. Mendoza, C. V. L., & Kleinschmidt, J. H. (2016). Defense for selective attacks in the iot with a distributed trust management scheme. In *IEEE 20th international symposium on consumer electronics* (pp. 1–2).
15. Chen, I. R., Guo, J., & Bao, F. (2016). Trust management for SOA-based IoT and its application to service composition. *IEEE Transactions on Services Computing, 9*(3), 482–495.
16. Bao, F., & Chen, R. (2012). *Trust management for the internet of things and its application to service composition* (pp. 1–6). Mobile and Multimedia Networks (WoWMoM): IEEE International Symposium on a World of Wireless.
17. Liu, W., Yin, L., Fang, B., & Yu, X. (2012). An efficient trust evaluation approach in attacker dominated networks in internet of things. *Future Information Technology, Application, and Service. Lecture Notes Electrical Engineering, 164,* 559–567.
18. Chen, D., Chang, G., Sun, D., Li, J., Jia, J., & Wang, X. (2011). TRM - IoT: A trust management model based on fuzzy reputation for internet of things. *Computer Science and Information Systems, 8*(4), 1207–1228.
19. Nitti, M., Girau, R., & Atzori, L. (2014). Trustworthiness management in the social internet of things. *IEEE Transactions on Knowledge and Data Engineering, 26*(5), 1253–1266.
20. Contiki Operating System for the Internet of Things. www.contiki-os.org

Carolina Veronica Lezama Mendoza was born in Nicaragua and earned a Master of Science in Information Engineering at the Federal University of ABC, Brazil (2014). Her areas of interest are mobile networks, Internet of Things and information security.



João Henrique Kleinschmidt is Computer Engineer and Master in Computer Science at the Pontifical Catholic University of Paraná in 2001 and 2004, respectively. He earned a Ph.D. in electrical engineering from the State University of Campinas in 2008. He is currently professor at the Federal University of ABC in Santo André-SP, Brazil. His research interests are computer networks, wireless communications, distributed systems and information security.