Implementing a Trust and Reputation Model for Robotic Sensor Networks

G. Tuna[1], S. M. Potirakis[2], G. Koulouras[3]
[1]*Department of Computer Programming, Trakya University, Edirne, Turkey*
[2]*Department of Electronics Engineering, Technological Education Institute of Piraeus, Aigaleo, Greece*
[3]*Department of Electronics Engineering, Technological Education Institute of Athens, Aigaleo, Greece*
gurkantuna@trakya.edu.tr

***Abstract*—Robotic sensor networks (RSNs) can be described as networks of devices equipped with communication, sensing and actuation capabilities. Successful implementations of RSNs require tackling challenging problems lying at the intersection of robotics, communication and perception. In addition, RSNs resemble human societies and emerging intelligent multi-agent systems in some respects. In these collaborative distributed systems, each node decides which to interact with and forms a network with other nodes in order to improve the quality of the decisions. In order to achieve this goal, trust and reputation models are of practical use. In this paper, a trust and reputation model for RNSs is proposed. Also, performance evaluations of the model in comparison with well-known models in the literature are given to prove its effectiveness. The results of the performance evaluations prove that the proposed model is successful in RSNs comprising of a large number of sensor nodes. In addition, the processing and memory requirements of the proposed model are moderate and the system runs effectively in systems with low processing power and limited main memory.**

***Index Terms*—Robotic sensor networks, security threats, trust and reputation model, comparative performance evaluations.**

I. INTRODUCTION

Robotic sensor networks (RSNs) are distributed systems comprised of mobile robots and sensors. The use of mobile robots offers new capabilities to wireless sensor networks (WSNs) which have wide applicability in different scientific, industrial, military, consumer and medical applications in real world implementations. Mobile robots can greatly reduce a number of problems related to issues including deployment, power management, failure detection, security and calibration which hinder the potential benefits of WSNs in practice. In RSNs, mobile robots which can take different roles such as sensing, routing and acting as data mules depending on the scenario carry sensors around an environment in order to detect physical phenomena and produce assessments. RSNs can be utilized for different purposes including environmental monitoring, intelligent agriculture, industrial control and monitoring, monitoring of machinery, military and security sensing, tracking of assets, management of supply chains, dynamic sensor placement, health monitoring, and wide area surveillance & reconnaissance on land, air or water.

RSNs bring several benefits which are accompanied by a number of significant risk factors and potential for abuse. Hence, how can users trust the information provided by the RSNs? In this respect, the main security goals of a RSN are to protect the RSN against all types of attacks including impersonation, fabrication, injection and modification of packets, node capturing, eavesdropping, and to address related issues such as privacy, accountability, availability, data authentication, data integrity and freshness. All these issues apply to traditional networks, too. But, due to resource constraints, open transmission medium and unattended deployment, they may have more severe consequences in RSNs.

In order to deal with the above-mentioned problems, trust and reputation models are of practical use. Regardless of the target platform a trust and reputation model (TRM) aims at and the type of threat, a good model should react against the threat and readapt itself as quick as possible [1], [2]. Overall, this paper presents different trust and/or reputation models which can be implemented in RSNs. Different trust and reputation models (TRMs) are compared in terms of applicability, practicability and effectiveness.

The remainder of this paper is organized as follows. Section II introduces different TRMs in the literature which were proposed for different systems and networks which include distributed multi-agent systems (DMASs), peer-to-peer (P2P) systems, mobile ad-hoc networks (MANETs) and WSNs. Section III introduces a TRM for RSNs. Performance evaluations of the models explained in Section II and III are given in Section IV. Finally, the paper is concluded in Section V.

II. RELATED WORK

In this section, we introduce various TRMs in the literature since RSNs exhibit the characteristics of different

Manuscript received December 13, 2012; accepted May 11, 2013.

systems at the same time. In DMASs, each agent may collaboratively select which agent to interact with [1]. In order to achieve the goals of DMASs without putting the systems at risk, many TRMs have been proposed. In [3], a reputation system for DMASs, Sporas, is proposed. In this system, reputation is computed recursively and each rating is assigned a weight. In the model proposed in [4], Regret, reputation is managed in a different way. In this model, there are three types of reputation: individual reputation, social reputation, ontological reputation. Direct interactions with an agent are used to build individual reputation. Experiences of other agents in the group to which the agent belonged are used to build social reputation [1]. On the other hand, when multiple aspects of reputation are combined, ontological reputation is built. In the reputation mechanism proposed in [5], AFRAS, the reputations of agents and the ratings of interactions are modelled by using fuzzy sets. In [6], a Bayesian network is used for computing the trust value among agents. The reputation system used in this model, MTrust, relies on a feedback submission algorithm. Other models proposed for multi-agent systems are given in [7]–[9].

Though P2P systems are very common in the distribution of information, they are open to several threats [2], [10]. The use of community based reputations is one of the ways which help estimating trustworthiness of peers. In order to deal with the threats in P2P systems, many models have been proposed. In [11], a model which relies on dynamic weights that represent the factors having influence on the trust, DWTrust, is proposed. In this way, the modelling and the computing of the trust are simplified. In [12], a model which relies on swarm intelligence, AntRep, is proposed. This model uses an ant system to build trust relationships. EigenTrust, a popular trust model for P2P systems, is proposed in [13]. Based on the history of uploads, each peer is assigned a unique trust value. In this way, total downloads of inauthentic files are reduced. As in other models, information coming from many sources is used to determine how trustworthy an entity is. In addition to this, pre-trusted entities are acceptable in EigenTrust. Therefore, EigenTrust is vulnerable to community structure and targeted attacks based on eigenvector centrality, since it ranks nodes close to the pre-trusted ones higher than the nodes further away. PeerTrust, a TRM which combines several key aspects of trust and reputation to develop a trust mechanism, is proposed in [14]. Using PeerTrust, peers can evaluate the trustworthiness of other peers and perform trusted interactions based on the history of interactions. Its strengths are its satisfactory outcomes and its context factor used to distinguish the trust given to a peer for different transactions. On the other hand, it measures the credibility of a peer without distinguishing between the confidence placed on a peer when providing a service and when giving recommendations about other peers.

In MANETs, countermeasures need to be implemented to deal with misbehaving nodes. In [15], a robust reputation system for both P2P systems and MANETs, RRS, is proposed. In this reputation system, each node maintains rating lists both for reputation and trust about others. At predefined intervals, reputation information is exchanged with the others. In [16], a decentralized trust model, PTM, is proposed. In this fuzzy logic based trust model, each node maintains a key pair which consists of available certificates, behavioural information and a list which categorizes users as trustworthy or untrustworthy ones.

Though WSNs bring several benefits to monitoring applications, control applications, and many others, they are open to a several types of security threats [17], [18]. The main goal of TRMs in WSNs is to provide information which permits nodes to identify which nodes are trustable. In addition, these models help coping with observable misbehaviour and minimizing the threats of inside attackers. In [19], a reputation and security model which is based on ant colony optimization, QDV, is proposed. In this model, a distance vector protocol detects malicious nodes in order to protect WSNs. When a node has more reputation, it is more reliable for communication. An agent based trust and reputation management model developed considering the limited resources of sensor nodes, ATRM, is proposed in [20]. This management model is executed locally to minimize overhead. Mobile agents running on nodes are responsible for the administration of the trust and reputation of their hosting nodes. BTRM-WSN, a trust model based on a bio-inspired algorithm, is proposed in [21]. Using BTRM-WSN, nodes can find the most trustworthy path which leads to the most reputable provider in a WSN. The model easily adapts to immediate changes in the topology.

Different from the target platforms of most TRMs, RSN implementations are generally real-world scenarios in which specific tasks need to be completed in real time or with a little delay. In addition, the requirements of the application scenarios and the inherent limitations of RSN nodes impose additional burden on the design of TRM models proposed for RSNs. Therefore, the use of complicated and time consuming TRM models for RSNs is questionable.

III. A TRUST AND REPUTATION MODEL FOR ROBOTIC SENSOR NETWORKS

Trust is important in the decision making processes of any systems including RSNs. When uncertainty is one of the factors in an environment, there is a need for a trust management system (TMS). Generally, TMSs are classified into two categories based on the approach as follows:

1. Credential based TMSs: In these systems, credential verification is used in order to establish trust and restrict access to resources according to previously defined policies;
2. Behaviour based TMSs: In these systems, agents trust other agents based on their past behaviour or experience, the concept of reputation. Thus, nodes can perform evaluations on the other agents based on these features.

To address the issues which decrease the robustness of RSNs against malicious attacks and rapidly changing topologies, a number of approaches can be used as follows:

- Creating and managing trust and reputation tables for all RSN nodes;
- Finding out misbehaving and/or faulty nodes and reporting them for exclusion;
- Using low-overhead cryptography for protecting the

authenticity and integrity of exchanged data;

– Applying watchdog mechanisms for monitoring the behaviour of surrounding RSN nodes;

– Using mechanisms for guaranteeing that RSN nodes comply with protocol rules.

In RSNs, nodes obtain the physical information of their surroundings, process the raw information, and finally communicate with other RSN nodes using wireless channels. Though all RSN nodes are battery-operated and mobile, they may have different computational capabilities. In RSNs, nodes can move to specific points when required. Hence, the topology changes whenever nodes move to other points. This is a rather problematic issue from the implementation point of view. The network model of the RSNs is determined by the organization of both the RSN nodes and the base station. In RSNs base stations are also mobile, but they can be static in some cases depending on the scenario. There may be no base station in some specific scenarios.

MANETs, WSNs and RSNs have common characteristics which create differing considerations compared to other systems when determining trust and reputation as follows:

– Self-organization: They are autonomous networks and do not have fixed infrastructures or centralized administrative nodes;

– End-to-end communication: Communication inside these networks generally requires packet forwarding for information to reach their destinations;

– A dynamic topology: Dynamically changing topologies require scalable and reliable security mechanisms;

– Limited bandwidth: Bandwidth limitations of nodes create considerable constraints.

On the other hand, RSNs differ from MANETs and WSNs since they have more processing power and energy resources.

Though mobility brings several advantages to traditional ad-hoc networks such as context aware deployment [22], continuous calibration, renewable energy which can be provided by mobile service robots, appropriate security measures must be taken in order to improve the security of the RSNs due to the inherent properties of RSNs which make them be prone to the surrounding environment and suffer from several types of attacks. Since the survival of RSNs depends on the trusting and cooperative nature of their nodes, establishing trusts between nodes is a must. Though sometimes security and trust are used interchangeably, they are different concepts; but they are tightly interdependent concepts. The definition of trust can be built upon reputation which is the opinion of one person about the other, of one agent about another agent, and by construct, of one RSN node about another RSN node. In RSNs, trust is derived from the reputation of a node. Based on the node's history of behaviour, the reputation is built over time and may reflect a negative or positive assessment as a result. In other words, while trust represents the opinion of a node of another node's reliability, honesty and capabilities based on its own experiences, reputation is the node's opinion of another node's reliability, honesty and capabilities based on recommendations received from other nodes.

In distributed RSN applications, while some nodes offer services, other nodes request these services. The requesting nodes need models which help selecting the best service providers according to certain criteria. Though each model has specific characteristics and particularities, most models share the same steps in order to complete transactions in a distributed system. The steps of TRMs designed for RSNs can be designed as similar to the steps of the well-known models in the literature such as [2], [23] and [24]. Except for mobility and longer node life provided by the internal batteries of mobile robots, RSNs exhibit the characteristics of WSNs and other distributed systems. Hence, after identifying the main steps of the well-known models, we have designed a model shown in Fig. 1.

The first step of the model gathers behavioural information about nodes in the system. The behavioural information obtained from many sources including direct experiences, neighbours and belonging groups or organizations is used to determine the absolute or relative trustworthiness of the nodes. The important point of this step is to take confidence levels into consideration when the information is provided by indirect sources [25].

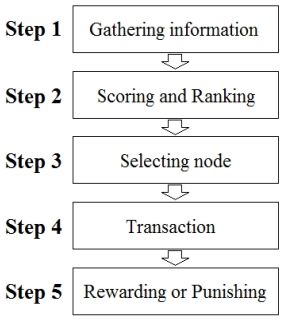

Fig. 1. The steps of general trust and/or reputation models for RSNs.

The information obtained from neighbouring nodes is a kind of second-hand information and the authenticity of its sources and the integrity of their contents cannot be guaranteed directly. Hence, a mechanism for assuring the correct management of the information is necessary [25], [26]. The second step is to compute a score for the nodes after collecting the history of transactions and weighing them. Bayesian networks, fuzzy logic, analytic expressions and algorithms are some of the methods used for computing the score [1], [23]. After scoring, a global ranking is done. This ranking is used to help deciding which node to interact with. Using the rankings obtained in Step 2, the node to interact with is decided in the third step. After node selection, the transaction is carried out between both nodes. Finally, after the transaction, the client node assesses the transaction. After the assessment, it rewards or punishes the servicing node.

Since behavioural information gathered in Step 1 is received from many sources in the network and it is propagated in one or multi hops depending on node locations, a complementary mechanism to minimize bandwidth usage is necessary. In order to optimize this distribution, we propose a strategy similar to the one proposed in [27]. The strategy relies on many factors including the resource constraints of the RSN nodes and the routing protocol used in the network. Behavioural

information distribution is critical since *scoring and ranking* step, Step 2, mainly depends on the behavioural information in addition to the personal evaluations of the nodes. There are different approaches to control the distribution of behavioural information as follows:

- In the most basic approach, RSN nodes only deliver to their immediate neighbours. To implement this approach, nodes maintain tables which include next hops for all routes.
- In an alternative approach, to propagate behavioural information, RSN nodes can use limited flooding in which the reports are allowed to travel a predetermined distance. In this way, the amount of additional network load is limited. This approach permits RSN nodes to contribute to behavioural information more than the basic approach.
- In a more systematic approach, RSN nodes process all messages in order to extract the routing nodes from the source nodes up to and including the next hop nodes. In this approach, routes are stored in a table. When a threat or good behaviour is observed by a node, the node examines the table in order to identify the source nodes which recently routed messages through this node. Then, the node informs the source nodes. This approach also minimizes the amount of additional network load.

Aggregating the behavioural information obtained from many sources is an important step of the proposed approach. Behavioural information can be aggregated by taking the reliability levels of nodes into consideration as given in (1). Reliability levels can be obtained from intrusion detection systems (IDSs) running on nodes or can be assigned dynamically based on the transaction history. While low values mean the existence of malicious activity or unreliability, high values exhibit reliability. In our system, we assign numerical values to nodes as listed in Table I. Our aggregation approach takes the distance to each RSN node delivering behavioural information report. In this way, behavioural information reports from RSN nodes located outside the neighbourhood of the receiving node can be accepted and the distance of these nodes can be considered during the evaluation of reports. Our approach also includes a report aging scheme which ensures that stale behavioural information is not used. The main steps of the algorithm used in the proposed model, TRM for RSNs, are explained as follows.

TRM for RSNs
Input: Behavioural information from nodes
1: Eliminate stale behavioural information using the report aging scheme
2: Compute a score for the nodes using (1) and the reliability levels in Table I.
3: Perform a global ranking
4: Select the node to work with
5: Perform transaction with the node
6: Assessing the node
Output: Punish or reward decision

$$RL_i(j) = \frac{\sum_{k=0}^{n}[(RL_i(k) \div RL_{req}) \times RL_k(j) \times CR_i(k) \times (HM_i - H_i(k))]}{n \times CM_i \times HM_i}, \quad (1)$$

where $RL_i(j)$ represents the reliability level of RSN node j observed at node i, n represents the number of nodes which report about node j, $RL_k(j)$ represents the reliability level which node k has of node j. RL_{req} represents the required reliability level for the delivery of current message. $CR_i(k)$ represents the remaining cycles used to diminish the contribution of older reports at node i for node k. CM_i represents the maximum number of cycles permitted at node i. $H_i(k)$ represents the distance, in hops, from node i to the reporting node k. HM_i represents the maximum distance which can be permitted for node i.

TABLE I. RELIABILITY LEVELS USED IN THE AGGREGATION APPROACH.

Value	Description
0	Malicious or compromised
1	Unknown reliability level
2	Low reliability
3	Medium reliability
4	High reliability
5	Very high reliability

IV. PERFORMANCE EVALUATIONS

Performance evaluations of the models have been carried out by using the simulator shown in Fig. 2, *Trust and Reputation Models Simulator for Robotic Sensor Networks*, developed with NetBeans IDE 6.9.1 in Java. Source codes of *TRMSim-WSN trust and reputation models simulator* [28] were used to develop our own Java-based simulator. TRMSim-WSN simulator was designed to simulate different trust and reputation models proposed for WSNs. Different from this simulator, the simulator we developed mainly aims at evaluating the model explained in Section III. But the models explained in Section II can be evaluated using our simulator. Similar to TRMSim-WSN, we have included oscillating server behaviour [23] and collusion threats in the simulator. If oscillating server behaviour option is selected, malicious servers become benevolent or vice versa after a predefined number of iterations. If collusion option is selected, then malicious servers form collusions among themselves. Malicious servers assign their maximum rating for other malicious servers and the minimum rating for benevolent servers [28].

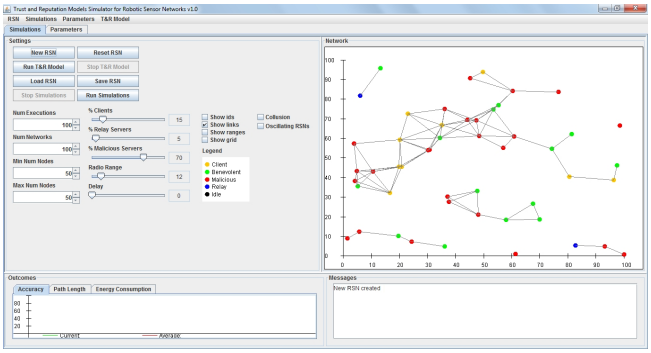

Fig. 2. Trust and reputation model simulator for RSNs.

The base part of simulator we developed is TRM_RSN Java class. This class contains several generic parameters. These generic parameters are provided by TRM_Param class. New TRMs can easily be added to the simulator by implementing subclasses of TRM_RSN and TRM_Param. TRM_RSN class also defines a set of public abstract methods including gatherInfo, score_and_Ranking,

performTran, rewardNode, punishNode in order to perform the steps shown in Fig. 1. These methods use arguments and return their values in objects. Each next step uses the return value of the previous step as it is illustrated in Fig. 1. However, in some TRMs, punishment and reward mechanisms are not used. Thus, depending on the TRM being implemented in the simulator, rewardNode and punishNode methods may not have any code.

In our first set of performance evaluations, we have computed the average satisfaction level by collecting the satisfaction of all clients belonging to the tested RSNs which comprised of 100 nodes. The total number of RSNs was set as 100. For each RSN, the number of clients was set as 15, of relay servers was set as 5, of malicious servers was set as 70, and of benevolent servers was set as 10. The model was executed on each RSN for 100 times. According to the results of our performance evaluations, we have observed that the average success of the proposed model, selecting the most trustworthy server (a benevolent server), was 85.92 %. The average number of hops required to reach the most trustworthy server was 5.41.

In our second set of performance evaluations, by using the parameters of the first evaluations, we have compared the average success of the model for different number of sensors. The notebook used for the simulation studies has an Intel Core I5 460M CPU and 8 GB main memory. Table II lists the results of the model in addition to CPU and memory usages. As listed in Table II, the success of the model has not changed considerably when we have changed the number of nodes. On the other hand, when we have increased the number of nodes, the CPU and memory usages of the simulator have increased considerably. Also, simulation time heavily depends on the total number of nodes. Hence, there is a trade-off between the total number of RSN nodes and the practical applicability of trust models.

TABLE II. THE RELATION BETWEEN THE NUMBER OF NODES AND THE ACCURACY OF THE MODEL.

Performance criteria/Number of RSN nodes	50	100	150	200
Accuracy (%)	86.76	85.92	84.93	84.67
Average CPU usage (%)	22.38	25.01	29.27	36.41
Average memory usage (MB)	74	136	228	319
Simulation duration (sec)	65	274	710	1594

In our third set of performance evaluations, we have evaluated the effect of CPU power on the processing time of the model. With this goal, the simulator has been run on two different systems. Table III lists the hardware specifications of the systems. Windows 7 Home Basic 64 bit operating system (OS) runs on System 1 and Windows 7 Starter OS 32 bit runs on System 2. Table IV lists the CPU power rankings of these systems which were performed by independent organizations. This table has been used to check the consistency of the results of the performance evaluations with the CPU power rankings. In this set of simulations, the number of RSNs was 100, and the number of nodes belonging to the RSNs was 125. For each RSN, the number of clients was 25, of relay servers was set as 5, of malicious servers was set as 75, and of benevolent servers was set as 20. The model was executed on each RSN for 100 times. Generally, the amount of processing power needed by an application determines CPU usage. The amount of memory space needed to hold a running application determines memory usage. Thus, no consistent relationship exists between CPU and memory usage. But, considering the results of the third set of performance evaluations listed in Table V, it can be seen that since CPU power cannot be used efficiently in most cases, subtasks of the simulator wait on the processing queue and result in higher memory usage on System 2 due to the data of the subtasks.

TABLE III. SPECIFICATIONS OF THE SYSTEMS.

System ID	**CPU**	**Main Memory**	**Graphics Card and Memory**
1	Intel Core I5 460M	8 GB	ATI HD5650 – 1 GB
2	Intel Atom N455	2 GB	Intel GMA3150 – 256 MB (shared)

TABLE IV. CPU FREQUENCIES AND PERFORMANCE COMPARISONS [29].

CPU	**CPU Frequency (MHz)**	**Test 1 – 3DMark06 CPU**	**Test 2 – Cinebench R10 Single**	**Test 3 – Cinebench R10 Multi**
Intel Core I5 460M	2530-2800	2923	3096	7022
Intel Atom N455	1660	477	547	856

TABLE V. THE RELATION BETWEEN THE SYSTEM PERFORMANCE AND THE EFFICIENCY OF THE MODEL.

Performance criteria	System 1	System 2
Accuracy (%)	85.96	86.21
Average CPU usage (%)	27.04	48.54
Average memory usage (MB)	167	312
Simulation duration (sec)	412	1344

In our fourth set of performance evaluations, we have evaluated the performance of the proposed model in comparison with the well-known models in the literature such as EigenTrust [13] and PeerTrust [14], though these models were originally proposed for P2P systems. These simulations have been run on System 1. In this set of simulations, the parameters of the third set of evaluation have been used. The number of RSNs was 100, and the number of nodes belonging to the RSNs was 125. For each RSN, the number of clients was 25, of relay servers was set as 5, of malicious servers was set as 75, and of benevolent servers was set as 20. The models were executed on each RSN for 100 times. As listed in Table VI, the success of the proposed model is higher than EigenTrust. Though the success of the proposed model is slightly less than PeerTrust, originally proposed for P2P systems, as listed in Table VI, it is computationally less intensive and requires less memory than EigenTrust and PeerTrust models. Thus, the proposed model is more suitable for RSN nodes with limited processing power and main memory.

TABLE VI. THE COMPARISON OF THE MODEL WITH EIGENTRUST AND PEERTRUST.

Performance criteria / Model	Proposed model	EigenTrust	PeerTrust
Accuracy (%)	85.96	85.81	86.12
Average CPU usage (%)	27.04	38.44	34.21
Average memory usage (MB)	167	190	185
Simulation duration (sec)	412	451	438

V. CONCLUSIONS

This paper presents a TRM for RSNs to cope with security threats and evaluates its performance through several simulation studies. Since almost all TRMs have common characteristics and RSNs exhibit the characteristics of different systems and networks, we have also described well-known models found in the literature. In addition, we have developed an interface to provide a common layer which can be used in the design of TRMs for heterogeneous environments consisting of mobile robots and sensor nodes and have implemented this interface in the simulator.

The performance evaluations presented here prove that the TRM proposed in this study achieves results similar to EigenTrust and PeerTrust TRM models in terms of accuracy. On the other hand, its memory requirement and processing overhead are lower than of those models. Another conclusion that resulted from the performance evaluations is that the success of the proposed model is independent of the specifications of target platforms.

We can finally conclude that the proposed model can be utilized to secure RSNs against several types of attacks and that it is a promising TRM for RSNs comprised of nodes with limited resources since it provides accurate results even in RSNs with a large number of nodes and requires less CPU and memory resources than the well-known models in the literature.

REFERENCES

[1] F. G. Marmol, G. M. Perez, "Towards Pre-Standardization of Trust and Reputation Models for Distributed and Heterogeneous Systems", *Computer Standards & Interfaces*, vol. 32, no. 4, pp. 185–196, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.csi.2010.01.003

[2] S. Marti, H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems", *Computer Networks*, vol. 50, no. 4, pp. 472–484, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.07.011

[3] G. Zacharia, P. Maes, "Trust management through reputation mechanisms", *Applied Artificial Intelligence*, vol. 14, pp. 881–907, 2000. [Online]. Available: http://dx.doi.org/10.1080/08839510050144868

[4] J. Sabater, C. Sierra, "REGRET: reputation in gregarious societies", in *Proc. of the Fifth Int. Conf. Autonomous Agents*, Montreal, Canada, 2001, pp. 194–195. [Online]. Available: http://dx.doi.org/10.1145/375735.376110

[5] J. Carbo, J. Molina, J. Davila, "Trust management through fuzzy reputation", *Int. Journal of Cooperative Information Systems*, vol. 12, pp. 135–155, 2003. [Online]. Available: http://dx.doi.org/10.1142/S0218843003000681

[6] S. Songsiri, "MTrust: a reputation-based trust model for a mobile agent system", *Lecture Notes in Computer Science*, vol. 4158, pp. 374–385, 2006. [Online]. Available: http://dx.doi.org/10.1007/11839569_36

[7] L. Mui, M. Mohtashemi, A. Halberstadt, "A computational model of trust and reputation", in *Proc. of the 35th Annual Hawaii Int. Conf. System Sciences*, Washington, USA, 2002, pp. 188–196. [Online]. Available: http://dx.doi.org/10.1109/HICSS.2002.994181

[8] S. Ramchurn, C. Sierra, L. Godo, N. Jennings, "A computational trust model for multi-agent interactions based on confidence and reputation", in *Proc. of the 6th Int. Workshop of Deception, Fraud and Trust in Agent Societies*, 2003, pp. 69–75.

[9] A. Abdul-Rahman, S. Hailes, "Supporting trust in virtual communities", in *Proc. of the 33rd Hawaii Int. Conf. System Sciences*, Hawaii, USA, 2000. [Online]. Available: http://dx.doi.org/10.1109/HICSS.2000.926814

[10] A. Josang, R. Ismail, C. Boyd, "A survey of trust and reputation systems for online service provision", *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2005.05.019

[11] C. Huang, H. Hu, Z. Wang, "A dynamic trust model based on feedback control mechanism for P2P applications", *Lecture Notes in Computer Science*, vol. 4158, pp. 312–321, 2006, [Online]. Available: http://dx.doi.org/10.1007/11839569_30

[12] W. Wang, G. Zeng, L. Yuan, "Ant-based reputation evidence distribution in P2P networks", in *Proc. of the Fifth Int. Conf. on Grid and Cooperative Computing*, Changsha, Hunan, China, 2006, pp. 129–132.

[13] S. Kamvar, M. Schlosser, H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks", in *Proc. of the Int. World Wide Web Conf. (WWW)*, Budapest, Hungary, 2003.

[14] L. Xiong, L. Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities", *IEEE Trans. on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2004.1318566

[15] S. Buchegger, J.Y. Le Boudec, "A robust reputation system for P2P and mobile Adhoc networks", in *Proc. of the Second Workshop on the Economics of Peer-to-Peer Systems*, Cambridge MA, USA, 2004.

[16] F. Almenarez, A. Marin, C. Campo, C. Garcia, "PTM: a pervasive trust management model for dynamic open environments, privacy and trust," in *Proc. of the First Workshop on Pervasive Security and Trust*, Boston, USA, 2004.

[17] M.C. Fernandez-Gago, R. Roman, J. Lopez, "A survey on the applicability of trust management systems for wireless sensor networks", in *Proc. of the Int. Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, 2007, pp. 25–30. [Online]. Available: http://dx.doi.org/10.1109/SECPERU.2007.3

[18] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, M. Cardei, *Reputation and Trust-based Systems for Ad Hoc and Sensor Networks*, in *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks*, A. Boukerche (ed.), Wiley, 2008. [Online]. Available: http://dx.doi.org/10.1002/9780470396384.ch13

[19] S. K. Dhurandher, S. Misra, M.S. Obaidat, N. Gupta, "An ant colony optimization approach for reputation and quality-of-service-based security in wireless sensor networks", *Security and Communication Networks*, vol. 2, no. 2, pp. 215–224, 2009. [Online]. Available: http://dx.doi.org/10.1002/sec.75

[20] A. Boukerche, L. Xu, K. El-Khatib, "Trust-based security for wireless ad hoc and sensor networks", *Computer Communications*, vol. 30, no. 11–12, pp. 2413–2427, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2007.04.022

[21] F. G. Marmol, G. M. Perez, "Providing trust in wireless sensor networks using a bio-inspired technique", in *Proc. of the networking and electronic commerce research conference*, Lake Garda, Italy, 2008.

[22] J. Gorecki, B. Miedzinski, H. Nouri, "Determination of Sensor Network Coverage using Probabilistic Approach", *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, no. 5, pp. 7–10, 2011.

[23] F. G. Marmol, G. M. Perez, "Security threats scenarios in trust and reputation models for distributed systems", *Computer & Security*, vol. 28, pp. 545–556, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2009.05.005

[24] Y. Sun, Y. Yang, "Trust Establishment in Distributed Networks: Analysis and Modeling", in *Proc. of the IEEE Int. Conf. on Communications IEEE ICC 2007*, Glasgow, Scotland, 2007.

[25] R. Roman, C. Fernandez-Gago, J. Lopez, H. H. Chen, *Trust and Reputation Systems for Wireless Sensor Networks*, in *Security and Privacy in Mobile and Wireless Networking*, Troubador Publishing, 2009, pp. 105–128.

[26] S. Japertas, G. Cincikas, R. Sestaviskas, "Company's Information and Telecommunication Networks Security Risk Assesment Algorithm", *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering)*, no. 5, pp. 33–36, 2012.

[27] Z. Liu, A. W. Joy, R. A. Thompson, "A dynamic trust model for mobile Ad Hoc networks", in *Proc. of the 10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems*, Suzhou, China, 2004, pp. 80–85.

[28] F. G. Marmol, G. M. Perez, "TRMSim-WSN, trust and reputation models simulator for wireless sensor networks", in *Proc. of the IEEE Int. Conf. on Communications*, Dresden, Germany, 2009.

[29] Mobile Processors Benchmark List. [Online]. Available: http://www.notebookcheck.net/Mobile-Processors-Benchmarklist.2436.0.html