ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

Ant colony trust based secure detection in wireless sensor networks

M.S Viji[1], Mr.R.Shankar[2]

Research Scholar, Dept. Of Computer Science, Chikkanna Govt. Arts College, Tirupur, India[1]

Assistant Professor in Computer Science, Chikkanna Govt. Arts College, Tirupur, India[2]

Abstract: Mobile wireless sensor networks (MWSNs) can simply be defined as a wireless sensor network (WSN) in which the sensor nodes are mobile. Most of the existing wireless sensor consumes more memory cost. The existing systems at most used on cryptography to improve packet security but this addresses only a part of the security problem without consideration for high energy key cost. The proposed algorithm to improve the routing security and monitoring activities of each node neighbors using status and trust worthy improves the security of WSNs and maximizes the lifetime for routing. The proposed algorithm using ant colony based best path choosing with secure packet routing. The bio inspired algorithm using ant colony system (ACS), where ants build paths satisfying positive conditions in a network graph. The simulation results show that the proposed model remains resilient to low or high percentages of pernicious service when the percentage of client node are greater than or equal 60%. Finally the simulation result performs well compare with existing one.

Keywords: Wireless sensor network, Ant colony optimization, Pheromone updating.

I. INTRODUCTION

Wireless sensor networks (WSNs) are vulnerable to several types of attacks including passive eavesdropping, jamming, compromising (capturing and reprogramming) of the sensor nodes, and insertion of malicious nodes into the network [1]–[3]. Widespread adoption of WSNs, particularity for mission-critical tasks, hinges on the development of strong protection mechanisms against such attacks [4]. Due to the scarcity of resources, traditional wireless network security solutions are not viable for WSNs. The life span of a sensor node is usually determined by its energy supply which is mostly expended for data processing and communication [5].

Moreover, size and cost constraints of the nodes limit their memory size and processing power. Therefore, security solutions which demand excessive processing, storage or communication overhead are not practical. In particular, due to their high computational complexity, public key ciphers are not suitable for WSNs.

Sensor Networks and related technologies have acquired considerable attention within the last 10 years. This is due to the truth that the technology is maturing and moving out of the purely research driven environment into commercial interests. WSNs serve to gather data and to monitor and detect events by providing coverage and message forwarding to base station. However, the inherent characteristics of a sensor network limit its performance and sensor nodes are supposed to be low-cost. An attacker can control a sensor node undetectably by physically exposing the node and an adversary can potentially insert faulty data or misbehavior to deceive the WSNs. Authentication mechanisms and cryptographic methods alone cannot be used to completely solve this problem because internal malicious nodes will have valid cryptographic keys to access the other nodes of the networks. Also conventional security methods cannot be used for WSNs due to power and processing limitations.

In addition to the node malicious raids, the nodes are also vulnerable to system faults for low-cost hardware of these nodes.

Recently, a new mechanism has been offered for WSNs security improvement. This mechanism relies on constructing trust systems through analysis of nodes observation about other nodes in the network. This article shows the last enhancement for WSNs by trust and reputation mechanisms found in literature. Research on the trust and reputation model is proposed for optimization in terms of security and scalability. This model is evaluated through applying security threats such as collusion and oscillating of malicious nodes in WSNs.

WSN are usually composed of a large number of these nodes which, together with their highly dynamic topology, may lead to some scalability problems. A number of research groups are deeply working on them since they have several interesting applications covering from military ones to environmental ones, passing through sanitary applications, domotics, Intelligent Transportation Systems (ITS), etc.

Proposed ACS is initially mainly designed for static networks, experiments demonstrate that the adaptations done to make it suitable for WSN lead to an accurate performance of the model. As we will see later, it allows a client to interact most of the times with a trustworthy server, rather than with a misbehaving one.

II. RELATED WORKS

Security is critical issue in a modern network system, although, often, one that the majority of the WSNs literature neglects to support minimizing energy consumption as the sole defining objective. The survey by [7] addresses a number of attacks that prove destructive to many essential WSN routing protocols. The security threats of WSN mainly contain external attacks and internal attacks. External attacks can be avoided by conventional encryption mechanism but it is not effective against internal attacks. As an important measure, reputation evaluation technique has an immediate effect on internal attacks [7].It has become an important measure to defend against internal attacks and it has received high concern. In recent years, an increasing number of researches have been conducted on the applying of reputation systems to sensor networks [8]. Meanwhile only [9] and [10] have concentrated on the use of reputation systems in WSN.

Probabilistic ciphers were also studied in [16] where a single cipher matrix is designed to minimize the error probability of AFC with a lower bound on the error probability of EFC. It was shown that it is possible to degrade the error probability of EFC significantly, and yet, achieve very low error probabilities for AFC. The design approach in [16] is ad hoc and results in a suboptimal solution for the cipher matrix. Security in distributed detection has also been investigated by many authors in the context of Byzantine attacks. Here an adversary inserts a number of malicious nodes into the network which deteriorate the detection performance by transmitting false data.

Moreover, some researchers have concentrated their effort in developing new trust and reputation models in the last decade. We have surveyed the related literature and have realized that most of those developers focused on describing their approaches. Many experiments presented and analyzed by researchers in order to prove the reliability of their proposals under certain conditions or circumstances. In [17] the use of Watchdog and Pathrater has suggested. Watchdog listens to the data transmission of the next node in the path to detect naughtiness. Pathrater keeps the ratings for other nodes and performs route selection by choosing routes that do not contain selfish nodes. However, the Watchdog mechanism needs high memory overhead to maintain the state information on the monitored nodes and the transmitted packets.

Researchers in [18] submitted a trust model to identify the trustworthiness of sensor nodes and to filter out the data transmitted by malicious nodes. In this model, researchers assume that every sensor node has knowledge of its own location coordinates, nodes are densely deployed and time is coincided. They evaluated trust in a conventional way, weighting the trust factors and there is no update of trust. Architecture based on reputation to create a network of autonomous sensors capable of detecting most kind of attacks and network failures using an anomaly detection system together with specification-based detection system have proposed in [19]. All this was created from the premise of designing a system that suit the characteristics of sensor networks and maintains the protocol as lightweight as possible to guarantee the autonomy of the nodes.

III. PROPOSED APPROACH

Proposed model is the first one in applying a bio-inspired technique such as ant colony system (ACS) to develop a trust and reputation model for WSN. Even more, it is the first one in providing the most trustworthy path (not only the most reputable node) leading to a specific sensor. Likewise, we have taken into consideration the important limitations found in WSN, so this system have tried to design a model as much lightweight, efficient, robust and scalable as possible.

Several types of WSNs can be found depending on what kind of nodes they are composed of. You can meet from a static WSN where nodes have a certain location, to a highly mobile one where nodes move everywhere. You can also find from a very restrictive WSN where all nodes remain most of the time asleep in an idle state, to another one comprising nodes provided with high performance features capable of process many requests per second and that are nearly always active.

A scenario where a WSN is composed of nodes with relatively high sensor activity. Without loss of generality, here consider some nodes requesting generic services and some nodes providing them. Assume that every node will only know its neighbors (that is, those nodes within its wireless range), and anything else about the whole topology of the net (at least at the early stages). Additionally, this topology is considered to be relatively highly dynamic, with many nodes entering or leaving the community.

Proposed model is aimed to help a node requesting a certain service to the network to find the most trustworthy route leading to a node providing the right requested service. A node (equally a path) can be considered untrustworthy either because it intentionally provides a fraudulent service or because it provides a wrong one due to hardware failures or performance deterioration.

Ant colony system (ACS) is a bio-inspired algorithm mainly used in optimization problems such as the travelling salesman problem or the quadratic assignation problem. It is based on the behavior of an ant colony in the nature. This algorithm is applied in those problems which can be modeled as graphs. Thus, a set of ants is launched and they start building paths fulfilling certain conditions. There are several concepts to be addressed here, such as how the pheromone traces are updated, how an ant decides which next node to transit to or how to measure the quality of each path found in order to keep the best one.

A. Bio-inspired trust and reputation model

Bio-inspired trust and reputation model for WSNs aimed to achieve to most trustworthy path leading to the most reputable node in a WSN offering a certain service. It is based on the bio-inspired algorithm of ant colony system but, due to the specific restrictions and limitations found in WSNs, the ACS cannot be directly applied. Some adaptations, therefore, have to be made. In our model, for instance, every node maintains a pheromone trace for each of its neighbors. These pheromone traces [0, 1] will determine the probability of ants choosing a certain route or another, and can be seen as the amount of trust given by a node to other one. The heuristic values [0, 1], however, are defined as the inverse of the delay transmission time between two nodes (or the inverse of the distance between them). The fact that every node controls its own pheromone traces and heuristic values, and no one else but it can modify them can become an important security threat.

Other issue that avoids the direct application of the ACS in this environment is the fact that while an ant is searching for the most reputable server providing a requested service, it could happen that some of the nodes that form the path followed by that ant become in accessible (either because they switch of or because they move out of the range of their previous sensor in the path). In that situation, the ant would be unable to come back to the client and it would get lost. In other words, when a client launches a set of ants, it has no guarantee at all that all of them are going to return and, of course, it cannot wait until all the launched ants came back in one iteration of the algorithm.

The first change we can appreciate is that the main now defined by a generic condition, which may be a certain number of iterations (like in the original algorithm) or it can even be a certain timeout. This definition will depend on the specific WSN this model is going to be applied to.

On the other hand, this algorithm consists of the following steps:

1. Every ant adds the first sensor to its solution, which is always the client they are departing from. Then each ant decides which next sensor to move to according to the transition rule and it is sent there.
2. Once every ant has left the client, this one waits until they come back. For every returned ant, the client compares its solution and keeps the best one. As explained before, in a WSN the client has no guarantee that all the ants that were launched are going to come back, so it just waits until a timeout expires or a certain percentage of all the ants has returned.
3. The best solution found by all or some of the ants issued in the current iteration is compared with the global best solution and swapped if it is appropriate.
4. Finally, a pheromone global updating is performed over the links belonging to the global best path.

Next we will describe in detail some features of our trust and reputation model for WSN, such as how to measure the quality of a path, how an ant decides which next sensor to travel towards, or when it should stop and return the current path. We will also explain how the pheromone updating is carried out while ants are building their routes as well as how a punishment is performed (in terms of pheromone evaporation) when the client interacts with a fraudulent server.

Algorithm steps

```
for It = 1 to Number  of  iterations do
for k = 1 to Number of ants do
Sk ← initial node
   for i = 2 to Number of nodes do
for k = 1 to Number of ants do
Sk←Sk [Transiti on Rule ($S_k\tau, \eta, \alpha, \beta, q_0$)
Pheromone local updating($S_k, \varphi, \tau_0$)
for k = 1 to Number of ants do
if (Q($S_k$) > Q(Current Best) ) then
Current Best←Sk
  if (Q(Current Best) > Q(Global Best) ) then
Global Best   Current Best
for i = 1 to Number of nodes do
Pheromone global updating(Global Best;Q(Global Best),$\rho$)
return Global Best
while (condition) do
for k = 1 to Number of ants do
 Sk←iinitial sensor (client)
Launch ant k
do
for every r e turned ant k do
 if (Q(Sk) > Q(Current Best) ) then
Current Best←Sk
while ( timeout does not expire ) and
 ( Number returned ant s < %Number of ants )
if (Q(Current Best) > Q(Global Best) ) then
Global Best   Current Best
Pheromone global updat ing(Global Best;Q(Global Best),$\rho$)
return Global Best
```

B. Path quality

Each time a launched ant returns to its client carrying a solution with it, that client has to assess the quality of that solution. Specifically the ant keeps a list of all the sensors belonging to the selected path, together with the pheromone traces of the links that join them.

According to this, the path quality computation can be done in the following way:

$$Q(S_k) = \frac{\bar{\tau}_k}{\sqrt{Length(S_k)}} \qquad (1)$$

Where $\bar{\tau}_k$ is the average pheromone of the path found by ant k and % Ak represents the percentage of ants that have selected the same solution as ant k.

C. Ants transition and stop condition

When an ant is travelling along the WSN searching for the most trustworthy route leading to the most reputable server it has to decide at each sensor which of its neighbors it has to move to. Every ant has also to decide whether to stop when it finds a server offering the requested service or if it should keep trying to find a more reputable one.

So let ant k be at sensor s in a certain moment of its searching. Several options can happen:

1. Sensor s offers the requested service.

(a) Sensor s has more neighbors not visited yet by ant k. The average pheromone of the path followed by ant k from the client until the sensor s is computed, $\bar{\tau}_k \epsilon$ [0, 1]. If $\bar{\tau}_k$ is greater than a certain transition threshold, TraTh ϵ [0, 1], then ant k stops and returns current solution with a probability defined by $\bar{\tau}_k$. Otherwise, if$\bar{\tau}_k \leq$ TraTh, ant k considers sensor s not enough reputable an keeps trying to search a better one.

(b) Sensor s has no more neighbors or all of them have been already visited by ant k. Ant k stops and returns current path.

2. Sensor s does not offer the requested service.

(a) Sensor s has more neighbors not visited yet by ant k. Ant k decides which next sensor to move to according to the expression shown in equation (2).

(b) Sensor s has no more neighbors or all of them have been already visited by ant k.

In this situation ant k has reached a dead end and has no more options than backtracking. That is, it has to follow the inverse route it has currently built until it is at a sensor which offers the requested service (and then stops and returns that path) or until

it reaches a sensor not offering the requested service but with more alternative paths not explored yet by ant k (and then keeps trying those routes). It could even happen that, while backtracking, ant k reached the client it belonged to. In that situation the whole WSN would have been explored but any server offering the requested service would have been found.

D. Pheromone updating

While ants are travelling across the WSN searching the most reputable server, they modify the pheromone traces they find. This modification helps next ants to decide which path is better to follow. Actually, there are two kind of updating: a local and a global one. The pheromone local updating is carried out by every ant each time it decides to move from one sensor to the next. Let ant k be at sensor s1. Then, applying the transition scheme explained in the previous section, it decides to move towards sensor s2 (which is a s1's neighbor). So, before being actually transmitted, it indicates sensor s1 that it has to modify its pheromone trace associated with sensor s2 in the following way:

$$\tau_{s1\,s2} = (1-\varphi).\tau_{s1\,s2} + \varphi.\Omega \qquad (2)$$

Where $\Omega = (1 + (1-\varphi).(1-\tau_{s1\,s2}\eta_{s1\,s2}).\tau_{s1\,s2}$ is the convergence value of $\tau_{s1\,s2}$ when t → 1, that is, is the pheromone trace value that would have that link after a lot of time if no other modification was carried out over it.

On the other hand, a pheromone global updating is performed over the best path found by all ants in each iteration of algorithm. This is done by sending an extra ant just to modify the pheromone traces of that route. And that modification is carried out using the next expression:

$$\begin{aligned}&\tau\_rs\\&= (1-\rho)\ \tau\_rs + \rho(1\\&+ \tau\_rs\ \eta\_rs\ Q(S\_(Global\_Best\ )\ )\ \tau\_rs \qquad (3)\end{aligned}$$

Therefore, the higher are the pheromone trace, the heuristic value, and the quality of the path, the higher is the additional pheromone contribution over the best route.

Finally, it is worth to mention how to initialize the pheromone traces. Their initial value IniPh ϵ [0,1] will condition some aspects of the model. Thus, if IniPh → 0, for instance, everybody would mistrust everyone at the beginning and it would be difficult to distinguish trustworthy sensors from malicious ones. However, if IniPh → 1 then everybody would trust everyone at the beginning and it would also be difficult to distinguish benevolent sensors form fraudulent ones. Therefore we decided that a good initialization value could be a random value close to IniPh, with IniPh → 0:5.

E. Server Selection

The most trustworthy path leading to the most reputable server, the client actually requests the desired service to that server. Then, depending on the goodness of the server, it will provide the same service it was offering, or another worse.

In this first stage we will consider only two possibilities. The server can be totally benevolent and provide the same service it was offering (so the client is fully satisfied), or it can be totally fraudulent and provide a completely different service than the one that was offered (having thus a fully unsatisfied client).

If the client is satisfied, a reward by means of additional pheromone contribution is done all along the selected path. The same expression used for pheromone global updating can be applied here as well. Nonetheless, if the client is not satisfied, a punishment, i.e., an evaporation of pheromone traces of the links belonging to the selected path, is carried out. And this punishment uses the following expression:

$$\tau_{rs} = (\tau_{rs} - \varphi.df_{rs}).\frac{S_{at}}{df_{rs}} \quad (4)$$

Where Sat ∈ [0,1] represents the satisfaction of the client with the received service and dfrs ∈ (0,1] is a distance factor of link e_{rs} computed as follows:

$$df_{rs} = \sqrt{\frac{df_{rs}}{L(S_k).(L(S_k) - d_{rs} + 1)}} \qquad (5)$$

drs ∈ {1, 2 ….L(Sk)} being d_{rs} the actual distance (number of hops) between sensor r and s, and $L(S_k)$ the length of the path found by ant k.

As it can be checked, having a punishing scheme like this, those edges which are closer to the client have slighter pheromone evaporation, and vice versa. Furthermore, all the links that fall into the malicious server are also punished. Otherwise ants could select it again through an alternative path, thinking it has become a benevolent sensor (which may not happen most of the times). Therefore, those edges have to be punished according to the next formula:

$$\tau_{rs} = (\tau_{rs} - \varphi)Sat \qquad (6)$$

The final equation to find the best path service server based on ant colony algorithm.

IV. Experimental Results

Proposed WSN where we are only interested on monitoring the behavior of sensors about just one service (or even if the WSN only provides one service), we could use this model without the problem of distinguishing a sensor's particular behavior for each provided service. But if we need a more resilient model, capable of dealing with multiple services, we could adopt the second version of WSN. In this one, every sensor has a pheromone trace for each one of its neighbors, and for each one of the services provided by the WSN.

Let be m the number of services available in the WSN, and let be ns the number of neighbors of sensor s. Then, s should manage and store m X n different pheromone traces. Obviously, this decision implies a bigger amount of stored information on each sensor but, on the other hand, it provides a more resilient trust and reputation model, since this is now able to distinguish each sensor as trustworthy or not, for each one of the services it offers.

Dealing with a WSN with high-resources sensors and where the security is a critical issue when applying for a service, we could make use of this second version of the model.

Java SWANS is used for simulation. All the experiments carried out consisted of 100 WSNs whose nodes were randomly distributed over an area of 100 square units. Of the nodes, requesting 100 times a certain service and applying a specific trust and/or reputation. Number of sensors used in the simulation is 50 and simulated for 100 executions. Another assumption in this simulation, every node only knows its neighbors within its RF range. Simulation parameters and default values used in the experiments are summarized in Table 1.

TABLE I
SIMULATION PARAMETER

Parameters	Value
Simulator	SWANS
Number of executions	100
Number of networks	100
Minimum number of sensors	50
Maximum number of sensors	Maximum number of sensors
Clients (%)	Variable
Malicious nodes (%)	Variable
delay between simulated networks	0
Radio range	12
Security threats used	Collusion and oscillating

V. PERFORMANCE COMPARISONS

Since our model has a strong basis on random or probabilistic decisions, we considered that it would be also quite interesting to take care about the standard deviation of that selection percentage of trustworthy servers. Finally, as a possible measure of the adaptability of our model specifically to WSN, we gathered as well the average path length of the solutions found by our model. As we mentioned before, in a environment with so many restrictions like WSN, the shorter path is always preferred since it supposes less consumption of sensors' resources.

PDR is the ratio of the number of data packets received by the destination node to the number of data packets sent by the source mobile node. It can be evaluated in terms of percentage (%). This parameter is also called "success rate of the protocols", and is described as follows:

$$\text{PDR} = ((\text{Send Packet no})/(\text{Receive packet no})) \times 100$$

Throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node.

$$\text{X} = \text{C}/\text{T}$$

Where X is the throughput, C is the number of requests that are accomplished by the system, and T denotes the total time of system observation.

Average end-to-end delay Average end-to-end delay signifies how long it will take a packet to travel from source to destination node. It includes delays due to route discovery, queuing, propagation delay and transfer time.

Protocols	Malicious Server								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
Optimal Probabilistic Encryption	92	86	83	78	72	69	64	58	55
Ant Based Trust	98	92	88	83	77	71	66	61	59

$$D_{end\,-end} = N(d_{trans} + d_{prop} + d_{proc}$$

Where delay end-end= end-to-end delay, dtrans= transmission delay, dprop= propagation delay,dproc= processing delay,dqueue= Queuing delay and N= number of links.

TABLE II SELECTION PERCENTAGE OF TRUSTWORTHY SERVERS

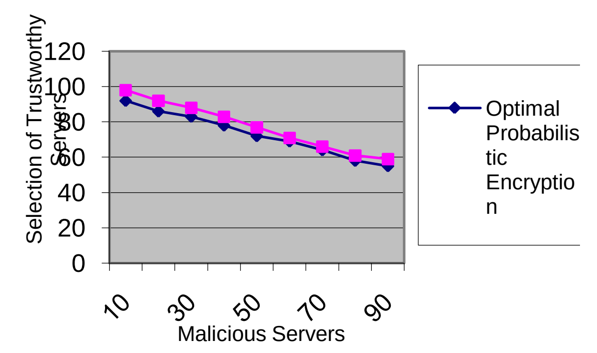

Fig. 1. Selection percentage of trustworthy servers.

Fig.1 shows they consider a trust and reputation model as acceptable (with a minimum quality level), in our opinion, the selection percentage of trustworthy servers should be greater or at least equal to 70%. A smaller percentage would result in a model with certain security deficiencies. And what is clear is that a selection percentage below the 50% means that the model is not useful at all.

TABLE III
AVERAGE PATH LENGTH LEADING TO TRUSTWORTHY SERVERS

Protocols	Malicious Server								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
Optimal Probabilistic Encryption	0.5	0.67	1.5	2.1	3.3	4.2	5.2	6.2	7.4
Ant Based Trust	0.8	1	2.3	2.8	3.8	4.7	5.8	8.1	9.1

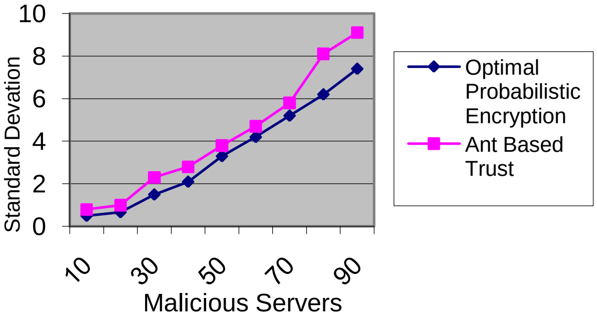

Fig. 2. Standard deviation of the selection percentage of trustworthy servers

TABLE IX
STANDARD DEVIATION OF THE SELECTION TRUSTWORTHY SERVERS

Protocols	**Malicious Server**								
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
Optimal Probabilistic Encryption	1.2	1.54	2.4	3.5	4.6	5.1	6.2	7.7	8.9
Ant Based Trust	1.8	1.92	3.1	4.23	5.43	6.43	7.45	8.4	9.3

One more time, differences between the several sizes tested for WSNs become distinguishable when the percentage of malicious servers is greater than or equal to 60%.our model is able to reach nearby trustworthy servers regardless the size of the network and the percentage of malicious servers. Although the smaller is the former and the greater is the latter, a larger path is found.

VI CONCLUSIONS AND FUTURE WORK

Proposed a Bio-inspired Trust and Reputation Model for WSNs, It is based on the Ant Colony System (ACS) and a complete description of its main features has been shown. Have seen how the pheromone traces deposited by ants help following ants to find the most trustworthy server through the most reputable path all over the network.

Specifically we have explained how the pheromone updating is carried out, as well as how to measure the quality of a path or how to punish or reward a server depending on its behavior. However, many future ways. For instance, a detailed description of some security threats that could be applied here can be an interesting issue. By managing each sensor its own pheromone traces, a malicious one could always assign the maximum value to other malicious neighbors or, equally, the minimum value to other benevolent ones. As future work, we need to apply experiments for the model using different network sizes and variable number of executions. Also, the balancing between the security and trust and reputation as per our scheme needs further investigation.

REFERENCES

[1] T. V. U. Kiran Kumar and B. Karthik, "Improving Network Life Time Using Static Cluster Routing for Wireless Sensor Networks ", *Indian Journal of Science and Technology*, Vol. 6, 2013, pp.4642-4647.

[2] Alkalbani, T. Mantoro, and A.O. Md Tap, "Improving the Lifetime of Wireless Sensor Networks Based on Routing Power Factors", Fourth International Conference on Networked Digital Technologies (NDT2012) , IEEE UAE Conference , Dubai, (UAE), 2012, pp.565-576.

[3] H. Chen, H. Wu, X. Zhou, and C. Gao, "Reputation-based Trust in Wireless Sensor Networks", International Conference on Multimedia and Ubiquitous Engineering (MUE'07), Seoul, (Korea), 2007, pp. 603-607.

[4] Josang, , R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision", Decision Support Systems, Vol. 43 No. 2, 2007, pp. 618-44.

[5] J. Sabater, and C. Sierra, "Review on Computational Trust and Reputation Models", *Artificial Intelligence Review*, Vol. 24, No. 1, 2005, pp. 33-60.

[6] J. Wang, Y. Liu, and Y. Jiao, "Building A Trusted Route In A Mobile Ad Hoc Network Considering Communication Reliability and Path Length", *Journal of Network and Computer Applications*, Volume 34, Issue 4, 2011, pp. 1138–1149.

[7] Karlof, and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp.113-27.

[8] Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei, "Reputation and Trust-Based Systems for Ad Hoc and Sensor Networks", *Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks*, 2007.

[9] Srinivasan, J. Teitelbaum, and J. Wu, "DRBTS: Distributed Reputation Based Beacon Trust System, Independable, Autonomic and Secure Computing", *2nd IEEE International Symposium on IEEE*, 2006. pp. 277–283.

[10] S. Ganeriwal, , L.K. Baizano, , M.B. Srivastava, "Reputation based Framework for High Integrity Sensor Networks", *ACM Transactions on Sensor Networks (TOSN)*, May 2008, Vol 4, Issue 3, pp. 15:1-15:37

[11] F. Almenarez, A. Marn, C. Campo, C. Garcfia, PTM: A pervasive trust management model for dynamic open environments, in: Privacy and Trust, First Workshop on Pervasive Security and Trust, Boston, USA, 2004.

[12] H. Chen, H. Wu, X. Zhou, C. Gao, Agent-based Trust Model in Wireless Sensor Networks, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 03 (2007) 119-124.

[13] S. Ganeriwal, M. B. Srivastava, Reputation-based framework for high integrity sensor networks, in: SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, ACM, New York, NY, USA, 2004, pp. 66{77.

[14] P. Michiardi, R. Molva, CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks, in: Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, Kluwer, B.V., Deventer, The Netherlands, 2002, pp. 107-121.

[15] Srinivasan, J. Teitelbaum, J. Wu, DRBTS: Distributed Reputationbased Beacon Trust System, in: DASC '06: Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, IEEE Computer Society,Washington, DC, USA, 2006, pp. 277-283.

[16] F. Li, Y. Wang, Routing in vehicular ad hoc networks: A survey, Vehicular Technology Magazine, IEEE 2 (2) (2007) 12-22.

[17] L. M. Gambardella, M. Dorigo, Solving symmetric and asymmetric TSPs by ant colonies, in: International Conference on Evolutionary Computation, 1996, pp. 622-627.

[18] L. Gambardella, E. Taillard, M. Dorigo, Ant Colonies for the QAP, Journal of the Operational Research Society 50 (1999) 167-176.

[19] [19] S. P. Marsh, Formalising trust as a computational concept, Ph.D. thesis, Department of Computing Science and Mathematics, University of Stirling (apr 1994).

[20] S. Marti, H. Garcia-Molina, Taxonomy of trust: Categorizing P2P reputation systems, Computer Networks 50 (4) (2006) 472-484.