IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727Volume 6, Issue 6 (Nov. - Dec. 2012), PP 21-26
www.iosrjournals.org

DTADA: Distributed Trusted Agent Based Detection Approach For Doline And Sensor Cloning Attacks In Wireless Sensor Networks

K.Arunasri[1], Mrs. K. Srimathi [2]
[1](K.Arunasri is currently doing her M.Tech at Sree Chaitanya College of Engineering, Thimmapoor, Karimnagar, AP, India.
[2](Mrs. K. Srimathi is currently working as Associate Professor, Sree Chaitanya College of Engineering, Thimmapoor, Karimnagar, AP, India)

Abstract: *Wireless sensor network (WSN) have diverse field of application, but it is extremely much prone to the security threats. This paper proposes a lightweight, fast, efficient and mobile agent technology based security solution against cloning attack and doline attack for wireless sensor networks (WSNs). WSN has a dynamic topology, irregular connectivity, and resource constrained device nodes. Researchers over the past years have confident the use of mobile agent to overcome these challenges. The proposed scheme is to defend next to cloning attack and doline attacks using mobile agents. Mobile agent is a program section which is self-controlling. They navigate from node to node not only transmitting data but also doing computation. They are effective paradigm for distributed applications, and particularly attractive in a dynamic network environment. This mechanism does not require more energy. Here we implement a simulation-based model of our solution to recover from cloning attack and sinkhole attack in a Wireless Sensor Network. contrast of communication overhead and cost were made between the proposed attack detection system using mobile agent against the security system in the absence of mobile agents.*

Keywords: *Wireless Sensor Network, Clone Attack, Doline attack, DTADA, DTADA4SCA, DTADA4DA*

I. Introduction

A Wireless Sensor Network (WSN) is a collection of sensors with limited resources that collaborate to achieve a general goal. WSNs can be deployed in harsh environments to fulfil both military and civil applications [1]. Due to their operating nature, they are frequently unattended, hence prone to different kinds of novel attacks. For instance, an adversary could listen in all network communications; further, an adversary could capture nodes obtain all the information stored therein —sensors are commonly assumed to be not tamper proof. Therefore, an adversary may duplicate captured sensors and deploy them in the network to launch a variety of spiteful activities. This attack is referred to as the sensor cloning attack [53], [11], [34]. Since a sensor cloning has legitimate information (code and cryptographic material), it may contribute in the network operations in the same way as a non-compromised node; hence cloned nodes can launch different attacks. A few have been described in the literature [3], [7]. For instance, a sensor cloning could create a black hole, initiate a wormhole attack [37] with a collaborating adversary, or inject false data or collective data in such a way to bias the final result [50]. Further, clones can leak data.

The threat of a sensor cloning attack can be characterized by two main points:

- Sensor cloning is considered totally honest by its neighbors. In fact, without global countermeasures, honest nodes cannot be conscious of the fact that they have a sensor cloning among its neighbors;
- to have a large amount of compromised nodes, the adversary does not require to compromise a high number of nodes. certainly, once a single node has been captured and compromised, the main cost of the attack has been continued. Making further clones of the similar node can be considered cheap.

In a sinkhole attack, the goal of an adversary is to lure nearly all the traffic from a particular area through a compromised node, creating a figurative sinkhole with the adversary at the middle. Because nodes on, or near, the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can allow many other attacks (selective forwarding, for example). Sinkhole attacks naturally work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an opponent could spoof or replay an advertisement for an extremely high quality route to a base station. One incentive for mounting a sinkhole attack is that it makes selective forwarding trivial. By make sure that all traffic in the targeted area flows through a compromised node, an opponent can selectively suppress or modify packets originating from any node in the area.

Two examples of doline attack are:

• Malicious node redirects with modified route sequence numbers. Here malicious node sends better sequence number to misguide that it is a fresh route.
• Malicious node redirects with customized hop count. Here spiteful node sends lesser hop count value to tell that this is direct path. In fact there is no such path exists.

To the best of our knowledge, with the exception of the protocol proposed in [45] and reviewed in the following, only centralized or local protocols contain proposed so far to cope with the sensor cloning attack. While centralized protocols contain a single point of failure and high communication cost, local protocols do not detect fake nodes that are distributed in different area of the network. In this work we look for a network self-healing mechanism, where nodes separately identify the occurrence of clones and exclude them from any further network action. In particular, this mechanism is designed to iterate as a "routine" event: It is designed for continuous iteration without significantly affecting the network performances, while get high sensor cloning detection rate. In this paper we study the desirable properties of distributed mechanisms for detection of node replication attack [17]. We also analyze the first protocol for distributed detection, proposed in [45], and show that this protocol is not completely satisfactory with respect to the above properties. Lastly, inspired by [45], we propose a new randomized, efficient, and distributed (RED) protocol for the detection of node replication attacks, and we show that our protocol does meet all the above cited necessities. We further provide analytical results when RED and its competitor face an adversary that selectively drops messages that could lead to sensor cloning detection. Finally, wide simulations of RED show that it is highly efficient as for communications, memory, and computations required and demonstrate improved attack detection probability (even when the adversary is allowed to selectively drop messages) when compared to other distributed protocols.

II. Related Work

Application of mobile agent computing model in WSN carries many advantages especially following ones [7-9]: (i) Decrease in energy consumption. Instead of data to be processed, agent is transmitted through network which can dramatically decrease quantity of data transmitted; (ii) Scalability. System performance without direct relationship with network scale, is supportive of balanced load; (iii) Reliability, which means the capability of overcoming the influence by unreliable network links through reaching the nodes accessed at time of establishing network links and Returning result after link recovered; (iv) Gradual computing accuracy. With the migration of mobile agent in network, computing result is required to become a gradually accurate. Once the requirement is met, mobile agent can return half-way with effect of energy saving. The proposed system uses mobile agent to detect cloning attack and to avoid doline attack in WSN.

One of the first solutions for the detection of sensor cloning attacks relies on a centralized Base Station (BS) [33]. In this solution, each node sends a list of its neighbors and their locations (that is the geographical coordinates of each node) to a BS. The same node ID in two lists with conflicting locations will result in a sensor cloning detection. Then, the BS revokes the clones. This solution has a number of drawbacks, such as the presence of a single point of failure (the BS) and high communication cost due to the large number of messages. Further, nodes close to the BS will be necessary to route much more messages than other nodes, hence limitation their operational life. Another central sensor cloning detection protocol has been recently proposed in [6]. This solution suppose that a random key pre-distribution security scheme is implemented in the sensor network. That is, every node is assigned a set of k symmetric keys, arbitrarily selected from a larger pool of keys [33]. For the detection, each node constructs a counting Bloom filter from the keys it uses for communication. Then, every node sends its own filter to the BS. From all the reports, the BS counts the number of times each key is used in the network. The keys used too frequently (above a threshold) are considered cloned and a corresponding revocation procedure is raised.

Other solutions rely on local detection. For example, in [9], [29], [33], [43] a voting mechanism is used within a neighborhood to agree on the legitimacy of a given node. However, this kind of a technique, applied to the problem of replica detection, fails to notice clones that are not within the similar neighborhood. As described in [45], a na¨ıve distributed solution for the detection of the node replication attack is Node-To-Network Broadcasting. In this solution every node floods the network with a message containing its location information and compares the received location information with that of its neighbors. If a neighbor sw of node sa collect a location claim that the similar node sa is in a position not coherent with the originally detected position of sa, this will result in a sensor cloning detection. However, this method is extremely energy consuming since it requires n flooding per iteration, where n is the number of nodes in the WSN. In the sybil attack [29], [43], a node claims multiple existing identities stolen from tainted nodes. Note that both the sybil and the sensor cloning attacks are based on uniqueness theft, however the two attacks are independent. The sybil attack can be capably addressed with mechanism based on RSSI [21] or with authentication based on the knowledge of a fixed key set [9], [14], [15], [25], [27]. Recent research threads cope with the more general problem of node compromise [19], [51], [47]. However, detecting node "misbehavior" via an approach that is rooted on techniques taken from intrusion detection systems [24] seems to require a higher overhead compared to sensor cloning detection. really, in present solutions detecting a misbehaving node implies observing, storing, and

processing a big amount of information. However, when mobility can be leveraged, security can develop incurring just limited overhead [13]. In particular, some beginning solutions start appearing in the literature that allow to recover sensor secrecy after node compromising [16], [23], [28], but these solutions do not cope with replica attacks.

Network overload is very high in many existing methods to detect doline attacks in WSN. And many existing approaches to detect doline attacks uses encryption and authentication mechanisms, it has encryption, decryption and key overhead. The proposed approach uses mobile agent to defend against doline attack to avoid all the above discussed disadvantages. Packet leashes [9] is based on geographical and temporal packet leashes. The use of geographical leashes supposes knowledge of the node location. The use of temporal leashes requires all nodes to have tightly synchronized clocks and demands computational power, which according to the authors, is further than the capability of sensors .SECTOR [14] is based on measurement of the time of flight of a message in a challenge–reply scheme. Such a scheme assumes that sensors are able to execute time measurements of nanosecond precision and, hence, this scheme need very accurate clocks at every sensor. In addition, distance estimates based on the time of flight are sensitive to distance-enlargement errors. Doline attack detection [15] finds a list of suspected nodes, and then carries out a network flow graph identifying a sink attack by observing data missing from an attacked area. The method is based on a central processing unit, which is not suitable in a wireless sensor network.

III. Distributed Trusted Agent Based Detection Approach

This system is designed to make every sensor aware of the location and identity of many nodes (Say n) so that Each neighbor of sensor A verifies the signature and checks the plausibility of Location of A. When a sensor finds a collision (2different location claims with the same ID), It broadcasts the two conflicting claims as evidence to revoke the replicas. And this system also makes every sensor aware of the entire network so that a valid sensor will not listen to the cheating information from malicious or compromised sensor which leads to doline attack. The above said two jobs are achieved with the help of mobile agents. Data routing algorithm tells how a sensor uses the global network information to route data packets. Thus the proposed system has two algorithms referred as DTADA4SCA (Distributed Trusted Agent based Detection Approach for Sensor Cloning Attack) and DTADA4DA (Distributed Trusted Agent based Detection Approach for Sensor Doline Attack), which are based on distributed trusted agent. DTADA4SCA is to detect cloned and unauthorized sensors. DTADA4DA is to notify how a sensor uses the global network information to route data packets by avoiding doline attack. The proposed system also has an Distributed Trusted Agent Routing Algorithm to route trusted agents which tells how does a distributed trusted agent gives the location and network information to nodes and visits every sensor.

i. Notations Used In DTADA:

$d_{s\to t}$	represents the transmission scope distance between two hop level sensors *s* and t and that can be measured as $d_{s\to t} = \dfrac{(tf - d)}{ms}$
' *tf* '	transmission frequency range of neighbor sensors *s* and t
' *d* '	distance between sensors s and t
' *ms* '	average sensor mobility speed
AAC	Agent Availability Counter tells how many times a trusted agent finds the particular sensor as a one hop neighbor or as a child sensor to the previous Node
ID	Agent id
ID_{tbl}	Agent Table contains ' *ID* ', ' *AAC* ' and location of the recent neighbor or child sensor
AVC_S	Is referred as agent visits counter, A counter at sensor s, which indicates the number times a trusted agent available as neighbor or parent to that's sensor. This also can claim as number of agent visits to sensor ' *s* '
$asc_{s\to t}$	Is an agent sequence counter at sensor s that indicates, the number of times sensor *t* is neighbor to ' *s* '
loc_i	Location of the sensor i
tbl_S	A table maintained by sensor ' *s* ' that contains 1. Transmission scope distance $d_{s\to t}$ to all other sensors, 2. Agent sequence counter *asc* 3. location of that sensor
sps_i	Signed position state of the sensor ' *i* '

ii. Conventions of the model proposed:

An agent can share ID_{tbl} with other trusted agents and sensors. The ID_{tbl} can update when an agent leaves a sensor.

If the transmission scope distance between any two sensors is more than the transmission range, there is a chance of a cloning attack.

There will be no updating if this entry is same as the location claim of sensor is carried by distributed trusted agent. Here more than one entry may be made only when a distributed trusted agent carries a different latest location claim for the same sensor.

IV. Distributed Trusted Agent Mobility Management Topology

In order to achieve this goal with the slightest overload, we put forward a slightest visited neighbor first algorithm to control the navigation of distributed trusted agents. An agent applies the algorithm to the information of sensor on which it currently resides, and decides its next destination. Every sensor has an information cache that agents can update with more recent values. Node access this shared cache whenever they require information about the network. When the agent reaches a sensor i, agent program do the following steps.

1. Step1: Updates tbl_i with ID_{tbl} if most recent updates found in ID_{tbl}. Vice versa updates ID_{tbl} with the information available at tbl_i, if information at tbl_i is latest than the information available at ID_{tbl}. Here in this case the updates are related to the number of times agent found this particular sensor as a one hop neighbor to the preceding or previous sensor in sequence of agent visits.
2. Step2: Agent gets the signed position state sps_i of that sensor i as follows

$sps_i = f_{(h)}(ID \mid loc_i)$

Then this compared with location of the sensor 'i' that cached in ID_{tbl}, if this comparison results similar, then no updates to ID_{tbl} required for sensor'i'. Otherwise checks the validity of loc_i such that the distance between location of sensor k, which is neighbor of sensor i, visited by agent and the current location of I must be less than the transmission scope distance $d_{i \to k}$. This validation can be done as follows

$diff(loc_k, loc_i) \le d_{i \to k}$

Here in the above equation '$diff(loc_k, loc_i)$' is using to measure the distance between sensors 'k' and 'i'.

If this validation process results true then 'ID_{tbl}' will be update with new 'loc_i'.

3. Step 3: Then the agent ID selects next sensor to be visited as follows:

Selects a neighbor sensor to the present sensor, which has been visited least number of times by agent. If duration from last visit to the selected neighbor node is less than the given visiting pause threshold δ then it selects next neighbor node in order that was visited least number of times.

4. Step4: And applies the step1 to step 3 on selected neighbor node of the present node.

V. Distributed Trusted Agent Based Detection Approach For Sensor Doline Attack (Dtada4da)

The process described here is used to route the data packets by avoiding the doline attack. Each sensor updates its information table tbl during the process of DTA-MMT. The data transmission from a sensor "' s ' to any other sensor 't' can be done with DTADA4DA. The process of DTADA4DA that helps to avoid doline attack explored below:

The source sensor s that aimed to send data to target sensor t, initially verifies the relation $rel_{s \to t}$ in its information table tbl_s which indicates the relation between 's' and 't'. The relation between s and t can be neighbor relation, child relation or no relation. The process of data packet routing under DTADA4DA explored in following steps:

Step1: Verifies the value of $rel_{s \to t}$ at information table tbl_s of sensor s. If $rel_{s \to t}$ indicates the any relation, then the data packets are sent to sensor t straight and hence routing ends. If $rel_{s \to t}$ indicates no relation then continues step2

Step2: Verifies the information table tbl_t of sensor t, If no relations found with any of the sensor available within the network region, then it will be confirmed that no sensor is able to reach sensor 't', hence routing process will be end since routing between sensor s and sensor t is not possible. If relations found between sensor t and other sensors $\{h_1, h_2, h_3, h_4, ...h_n\}$ then routing process proceeds to step 3.

Step 3: if $rel_{t \to h_i}$ representing a relation for sensors $\{h_1, h_2, h_3, h_4, ...h_n\}$, for each sensor h_i performs the step 4. If all sensors in list $\{h_1, h_2, h_3, h_4, ...h_n\}$ traversed then go to step 5.

Step 4: verifies the relation, if relation found go to step 6.

Step 5: perform step2, step3 and step4 for each sensor h_i as sensor '*t*' in the list $\{h_1, h_2, h_3, h_4, ...h_n\}$.

Step 6: Trace back the relations that leads to traverse from sensor *t* to sensor *s*, then perform data packet routing without doline attack between sensor *s* and *t*.

VI. Performance Analysis

When a distributed trusted agent is attempting to connect to a wireless sensor sensor which is in the sleep mode, the connection cannot be recognized. Here, the goal is to decide an optimal sleep cycle in one duty cycle. One duty cycle includes one sleep stage of s and one active period of (T-s). The probability p that the wireless sensor sensor cannot be connected is

$$p = s/T$$

If we have congestion probability as p and (T-s) =2Δ, we have

$$T = (2\Delta) / (1-p)$$

And

$$s = (2p\Delta) / (1-p)$$

Now, we need to calculate the power needed for a distributed trusted agent passing through set of nodes. Let P_S are being the power consumed at sleep mode, and P_a be the dynamic mode. Perceptibly, there are the following two cases in each responsibility cycle:

1. A distributed trusted agent at the target sensor has attempted all n nodes one after another in one duty cycle T. In this case, all nodes have a usual duty cycle with one sleep mode and one active mode. The energy in the normal duty cycle E normal is

$$Enormal = n\left(sPs + (T-s)\,Pa\right) = 2n\Delta\left(pPs + (1-p)\,Pa\right) / (1-p)$$

2. A distributed trusted agent at a wireless sensor has attempted remaining servers in one duty cycle T. In this case, all wireless nodes apart from one sensor (the one with the distributed trusted agent) have a regular duty. The sensor with the distributed trusted agent has to be in the active mode during the whole duty cycle in order to avoid loss of the distributed trusted agent. Hence, the energy in the special duty cycle Especial is

Especial= (n-1) (sPs + (T-s) Pa) + TPa = 2(n-1) Δ (pPs + (1-p) Pa) / (1-p) + (2Δ) Pa / (1-p)

For a distributed trusted agent to successfully obtain sensors' information, we can have r1 normal duty cycles and r2 special duty cycles, and the total energy required is Etotal= (r1+1) Enormal+ (r2-1) Especial.

Here, we put in one to r1 and minus one from r2 because, in the first special duty cycle, the first sensor accepts the distributed trusted agent and still operates in the regular duty cycle.

VII. Simulation Results

The proposed work was simulated using MXML and actionscript. The simulation parameters are explored in table 1.

TABLE 1: SIMULATION PARAMETER SETTING

Parameter	Value
Network scale	200m x 200m
No. of sensor nodes	25~400
Energy level	0~64
Distributed trusted agent code size	500 bytes
Bytes accumulated by the distributed trusted agent at each sensor sensor	100 bytes
Distributed trusted agent execution time at each sensor	50 ms
Distributed trusted agent instantiation delay	10 ms

Simulation parameter location is given in table 1. Figure 1 shows how is the average energy at nodes decreased for increased time because of storage of global information matrix at each sensor. Figure 2 shows the relationship between probabilities of doline detection with the number of nodes.

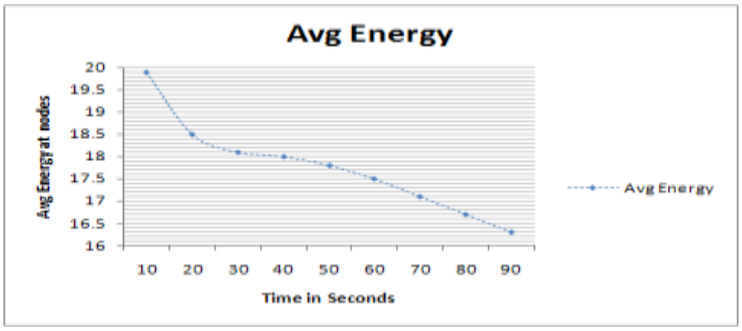

Figure 1: Average energy at nodes Vs Time

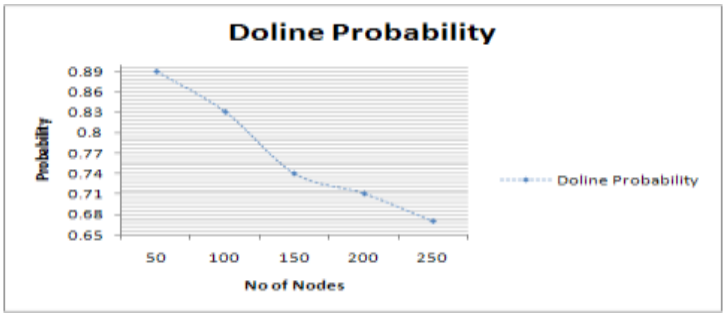

Figure 2: Probability of doline attack detection (y axis) with no. of nodes(x axis)

This paper doesn't show the simulation result of comparison between the communication overhead with an average probability of detection of the above said detection method using distributed trusted agent with the existing detection methods without using distributed trusted agent. We hope that this will motivate us to do it in future.

VIII. Conclusion

In this paper we propose a distributed trusted agent based detection approach to make sensor locations be learnt by other nodes with very less communication operating cost. Also this approach is used to make available necessary knowledge to every sensor in a Wireless Sensor Network not to believe the false path so that the doline attack can be shunned at influenced degree. The performance of the proposed approach has been examined through simulations.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, vol. 38, 2002.

[2] H. Chan, A. Perrig, and D. Song, "Random key redistribution schemes for sensor networks," in Proceedings of the 2003Symposium on Security and Privacy. Los Alamitos, CA: IEEE Computer Society, May 11–142003, pp. 197–215.

[3] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in Proceedings of the third international symposium on Information processing in sensor networks (IPSN-04).New York: ACM Press, Apr. 26–27 2004, pp.259–268.

[4] Zhang Yuyong, Jingde. Mobile Agent Technology. Beijing, Tsinghua University Press, 2003.

[5] Zhu Miaoliang, Qiuyu. Mobile Agent System. Journal of Computer Research and Development, 2001, 38(1): 16-25.

[6] G. Sladic , M. Vidakovic and Z. Konjovic Agent based system for network availability and vulnerability monitoring 2011 IEEE 9thInternational Symposium on Intelligent Systems and Informatics •September 8-10, 2011, Subotica, Serbia.

[7] L. Tong, Q. Zhao, S. Adireddy. Sensor Networks with Mobile Agents. IEEE Military Communications Conference, Boston, MA, USA,2003:688-693.

[8] M. Ketel, N. Dogan, A. Homaifar. Distributed Sensor Networks Based on Mobile Agents Paradigm. Proc. 37th Southeastern Symposium on System Theory, Tuskegee, AL, USA, 2005: 411-414.

[9] H. Qi, Y. Xu, X. Wang. Mobile-Agent-based Collaborative Signal and Information Processing in Sensor Networks. Proc. IEEE, 2003,91(8):1172-1183.

[10] B. Parno, A. Perrig, and V. D. Gligor, "Distributed detection of sensor replication attacks in sensor networks," in IEEE Symposium on Security and Privacy. IEEE Computer Society, 2005, pp. 49–International Conference on Computational Techniques and ArtificialIntelligence(ICCTAI'2012)Penang,Malaysia30363.Online].Available:http://doi.ieeecomputersociety.org/10.1109/SP.2005.8

[11] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "Arandomized, efficient, and distributed protocol for the detection of sensor replication attacks in wireless sensor networks," in Proc.ACM MobiHoc, 2007, pp. 80–89.

[12] B. Zhu, V. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of sensor replication attacks in sensor networks, "in Proc. 23rdAnn. Computer Security Applications Conference(ACSAC '07), Dec.2007, pp. 257–267.

[13] Y. Hu, A. Perrig, and D. Johnson, Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks, in: Proc. Of Infocom2003. San Francisco, CA, USA, April 2003.

[14] S. Capkun, L. Buttyan, J. Hubaux, SECTOR: Secure Traking of Node Encounters in Muti-hop Wireless Networks, in: proc. Of SASN 2003.Fairfax, Virginia, October 2003.

[15] E. C. H Ngai, J. Liu and M R. Lyu, "On the intruder Detection for Sinkhole Attack in Wireless Sensor Networks," Proc. IEEE ICC, 2006.