International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 5, Issue 3 (September '2019)

ANALYSIS OF TRUST MANAGEMENT IN UNATTENDED WIRELESS SENSOR NETWORKS

S.Kaviya,
M.Phil Research Scholar,
PG &Research Department of Computer Science,
PGP College of Arts & Science,
Namakkal,Tamilnadu,India.

K. Ranjith Singh,
Assistant professor,
PG &Research Department of Computer Science,
PGP College of Arts & Science,
Namakkal,Tamilnadu,India.

Abstract: An Unattended Wireless Sensor Network (UWSN) is a type of sensor network where a trusted sink visits each node periodically to collect the data. Due to offline nature of this network, every node has to secure the sensed data until next visit of the sink i.e. until data is being sent to the sink. UWSNs are operating in hostile environment where the goal of an attacker is to prevent targeted data from ever reaching the sink or to send corrupted data to the sink. So, in a UWSN data confidentiality and data authenticity are required. Besides we have to take care of survivability of sensed data. In this paper presented a scheme that provides authenticity, confidentiality and survivability of sensed data in efficient manner. All these issues were not addressed together in any of the previous scheme for securing UWSNs. Trust management systems have been recently introduced as a security mechanism in UWSN Here trust means the confidence of an entity on another entity based on the expectation that the other entity will perform a particular action important to the trustee. CONFIDANT [24] protocol is a reputation system that has been applied to WSN. In this paper described the Directed Diffusion routing protocol, different threats and attacks on UWSN, overview of trust and different trust models in UWSN and attacks on Directed Diffusion. Our trust model is introduced that works on directed diffusion.

Keywords: Unattended Wireless Sensor Network, Directed Diffusion routing protocol, Trust Management, Security

I. INTRODUCTION

Unattended Wireless Sensor Networks (UWSN) [1] (Fig. 1) is an hybrid type of WSN where data sensed by the sensors is not collected continuously by the sink. Data has to be secured by every node until the next visit of the mobile sink. This inability to communicate with sink might be for reasons such as: limited transmission ranges, power constraints or signal propagation problems.

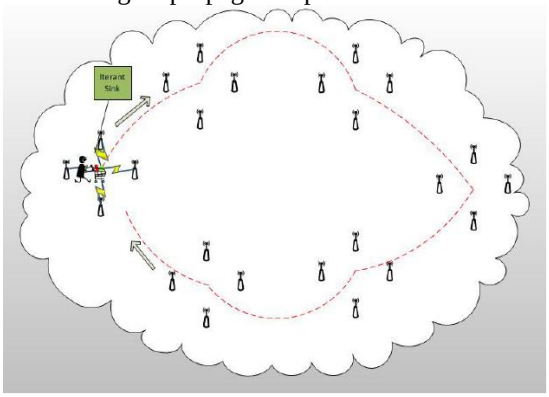

Figure 1: Unattended Wireless Sensor Network with Mobile Sink collecting the data

The concept of UWSNs with a mobile sink looks realistic if we consider the environments where the sensing field is too far from the base station and sending data through intermediate nodes may result in weakening the security (e.g., intermediate nodes may modify the data) or increase the energy consumption of the nodes close to the base station. In normal multi-hop Wireless Sensor Networks, power of the nodes placed near the sink will be depleted earlier than the other nodes. This is because all the nodes have to transmit the data to the sink through the nodes placed near the sink. An UWSN can be used to save the battery of these nodes and as a result increase the lifetime of the network Unattended environments as mentioned in [2] include sensor networks for monitoring sound and vibration produced by troop movement, airborne sensor networks for tracking enemy aircrafts, LANdroids [3] which retain information until soldiers move close to the network, sensor networks for monitoring nuclear emissions, national parks for firearm discharge and illicit cultivation, etc.

While dealing with security in UWSNs, we mainly focus on achieving some or all of the following security goals:

- **Forward Security:** The compromise of a secret in one round should not lead to the compromise of the secrets in the rounds preceding compromise. Forward security is critical in UWSNs so that even if the adversary can compromise the current key it is infeasible for it to generate the previous keys using the current key. The adversary cannot even forge the authentication tags for the data generated and authenticated before compromise.
- **Backward Security:** The compromise of a secret at any time should not lead to the compromise of the secrets to be used in the future. If an adversary obtains the current status of the node, it cannot decrypt the data generated and encrypted after compromise. Adversary should not able to forge the authentication tags for the data generated and authenticated after compromise.
- **Data Confidentiality:** Confidentiality ensures that the information is inaccessible to unauthorized users. In UWSNs, data in the sensors should be encrypted in a way that it can only be read by the sink. In some

scenarios, data from the nodes will not be sent to the sink in a single hop. Sink visits a point for data collection which can be few hops away from the node and the data has to be sent through other nodes. In these cases the intermediate nodes should not be able to read the transmitted information. Data confidentiality also prevents the read only adversary from reading the stored data in the compromised node's memory.

- **Data Integrity:** Data integrity protects against unauthorized alteration of the data. Data integrity can be achieved only if the network has the ability to detect the manipulations done to the data by unauthorized parties, i.e., insertion, substitution and deletion. Data integrity protects against the \search and replace" and \search and erase" types of adversaries.
- **Data Authentication:** Authentication applies to both nodes and data. It ensures the identity of the node with which it is communicating i.e., the two communicating parties can identify each other. Information delivered through the network should be authenticated with respect to the generation time; date of origin, origin etc., and Data origin authentication also provides data integrity as the message modification can be detected.

II. REVIEW OF LITERATURE

Gligor, *et al.* [4] the low-cost, off-the-shelf hardware components in unshielded sensor-network nodes leave them vulnerable to compromise. With little effort, an adversary may capture nodes, analyse and replicate them, and surreptitiously insert these replicas at strategic locations within the network. Such attacks may have severe consequences; they may allow the adversary to corrupt network data or even disconnect significant parts of the network. Previous node replication detection schemes depend primarily on centralized mechanisms with single points of failure, or on neighbourhood voting protocols that fail to detect distributed replications. Randomized multicast distributes node location information to randomly-selected witnesses, exploiting the birthday paradox to detect replicated nodes, while line selected multicast uses the topology of the network to detect replication. Both algorithms provide globally-aware, distributed node-replica detection, and line-selected multicast displays particularly strong performance characteristics. We show that emergent algorithms represent a promising new approach to sensor network security; moreover, our results naturally extend to other classes of networks in which nodes can be captured, replicated and re-inserted by an adversary.

Wehrle, *et al.* [5] Node compromise is a serious threat to wireless sensor networks deployed in unattended and hostile environments. To mitigate the impact of compromised nodes, we propose a suite of location-based compromise-tolerant security mechanisms. Based on a new cryptographic concept called pairing, we propose the notion of location-based keys (LBKs) [18][19] by binding private keys of individual nodes to both their IDs and geographic locations. We then develop an LBK-based neighbourhood authentication scheme to localize the impact of compromised nodes to their vicinity. We also present efficient approaches to establish a shared key between any two network nodes. In ontrast to previous key establishment solutions, our approaches feature nearly perfect resilience to node compromise, low communication and computation overhead, low memory requirements, and high network scalability. Moreover, we demonstrate the efficacy of LBKs in counteracting several notorious attacks against sensor networks such as the Sybil attack, the identity replication attack, and wormhole and sinkhole attacks. Finally, we propose a location-based threshold endorsement scheme, called LTE, to thwart the infamous bogus data injection attack, in which adversaries inject lots of bogus data into the network. The utility of LTE in achieving remarkable energy savings is validated by detailed performance evaluation.

Ferreria *et al.* [6] proposed the Break-the-Glass Role-Based Access Control (BTGRBAC) model based on the RBAC model. The main idea of this model is to gather necessary information from end users with their collaboration for a usable access control policy that can perform BTG action in emergency situations. The Break-The- Glass (BTG) rule allows users to have emergency and urgent access to the system when a normal authentication does not perform or work properly. Ferreria et al. introduced BTG rules in order to override access policy whilst providing non-repudiation mechanisms for its usage. In a real environment, unanticipated situations may occur because it is impossible to predict all access permissions in advance for all situations. BTG extension is used for emergency and important cases whenever a user wants to access data urgently and immediately. When the user tries to perform BTG actions, the system will ask him or her if he or she actually wants to perform that action on a specific object. If the user answers affirmatively, the system will activate the BTG operation and trigger the associated obligations such as alarms and log file. The BTGRBAC model has made the system much more flexible than normal RBAC but one of the disadvantages is that human processes are needed in order to define additional role and policy regarding BTG actions.

Wang *et al.* [7] proposed an access control model based on ECC. The main objective of the proposed model is to use an ECC scheme for granting user access rights to the collected data. Different users may have different levels of data access due to restrictions of access implied by the data confidentiality and privacy. ECC is used in key distribution and sharing information between the users and in a Key Distribution Centre (KDC). In this approach, the KDC is responsible for generating all security primitives such as random numbers, access lists and hash functions, and maintains a user list with associated user identifications. The users have to request access permission from the KDC. Access lists, which comprise user identity, group identity and user privilege mask, define the user's access privileges. The user access privilege mask consists of a number of bits and each bit represents a specific information or service. Therefore, users who possess the same mask and access privileges are put in the same group.

III. STRUCTURE OF TRUST MODEL

In wireless sensor networks, trust model plays an important role in identifying misbehavior nodes and providing collaboration among trustworthy nodes. It improves the

lifetime of networks that inspire expectations among future interactions. The model is capable of capturing and distributing feedbacks about current interactions among nodes and stores the trust information for future. It also uses feedback to guide trust decisions. Depending on trust information stored, it may be classified as centralized, distributed and hybrid model [8] and given in Figure 2. Centralized trust model consists of a single globally trusted server that determines the trust values of every node in the network. In distributed trust model, every node locally calculates the trust values of all other nodes in the network that increases the computational cost. Also each node needs to maintain an up-to-date record about the trust values of the entire networking in the form of a table. Hybrid trust model contains the properties of both centralized as well as distributed trust management approaches. The main objective of this approach is to reduce the cost associated with trust evaluation as compared to distributed approaches. This scheme is used with clustering schemes in which cluster head acts as a central server for that cluster. In the certificate-based trust model, trust is mainly based on the provision of a valid certificate assigned to a target node by a centralized certification authority or by other trusted issuer. In the behavior-based trust model, an entity calculates the trust values by continuous direct or indirect monitoring of other nodes

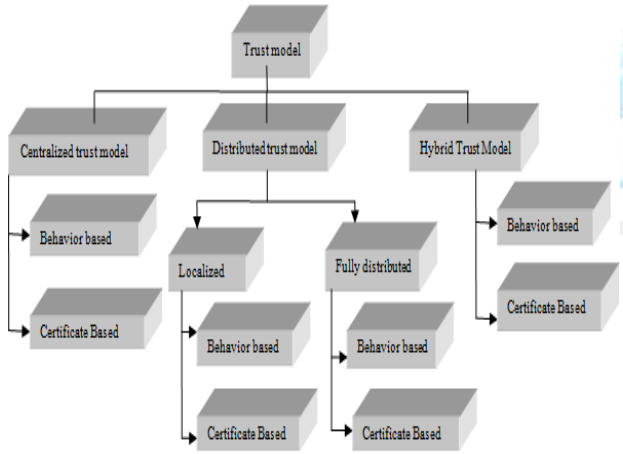

Figure 2: Structure of trust model

IV. DIRECTED DIFFUSION

Directed diffusion is a data centric routing protocol for data gathering in unattended wireless sensor networks proposed by Intanagonwiwat et. al. [1] Data generated by sensor nodes is named by attribute-value pairs. The sink requests data by broadcasting interests for named data. The interest is broadcasted by each node to its neighbors and thus it diffuses through the network. As the interest is propagated throughout the network, each node sets up gradient towards the nodes from it received the interest. A gradient can be thought of as a replying link pointing towards the neighboring node from which interests are received. Interest cache is maintained at each node. Sensor nodes detecting an event search their interest cache for an entry matching the interest. The source then sends data to each neighbor for whom it has a gradient. A neighbor node receiving data searches its interest cache for an entry matching the data. If no such entry is found then the data message is dropped, otherwise it is forwarded to the nodes for which there exists a gradient. In this way data is sent to the sink. Once, the sink starts receiving these data messages, it reinforces a particular neighbor in order to draw data at higher rates. The reinforced node similarly reinforces one of its neighbors to draw events at higher rates. Link failures as well as node failures can lead to data loss as well as reduced rate of data. When reduced rate of data is received from a reinforced node, it can be negatively reinforced and another neighbor providing data at a higher rate can be positively reinforced. Positive and negative reinforcement can be triggered by the sink as well as the other nodes. Directed diffusion has achieved significant energy-efficiency over flooding and omniscient multicast schemes. However, directed diffusion cannot be applied to applications requiring continuous monitoring and data delivery to the base station (e.g. environmental monitoring).

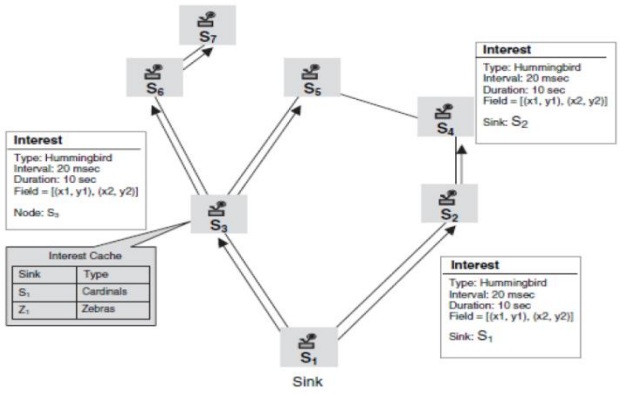

Figure 3: Interest Propagation

Figure 4: Gradient Setup

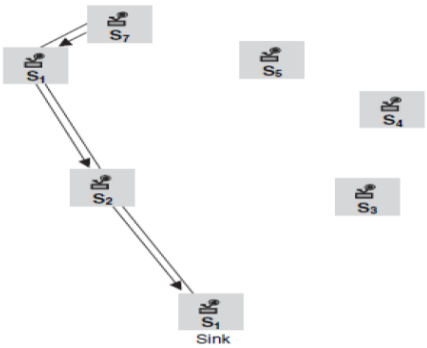

Figure 5: Data Delivery along a Reinforced Path

Attacks on Directed Diffusion: Since interests are flooded throughout the network in directed diffusion, it is difficult for the attacker to prevent them from reaching the sources. But when sources generate data matching the interests and forward them accordingly, an adversary can carry out the following attacks:

- **Suppression:** An attacker can spoof negative reinforcements to achieve flow suppression and perform denial-of-service attack.
- **Cloning:** An adversary receiving interest from a legitimate base station can replay the interest listing itself as a base station. Thus all events satisfying the interest would be sent both to the legitimate base station as well as the adversary. In this way, the adversary can eavesdrop on the information.
- **Path influence:** An attacker can influence the path taken by a data flow by spoofing positive and negative reinforcement. It can positively reinforce a weak link and negatively reinforce a strong link which was delivering data at a good rate thus causing disruption in data flow.
- **Selective forwarding and data tampering:** If an attacker is present in the path of flow of data, it can selectively forward and drop packets and also can tamper with the received data.

4.1 Directed Diffusion using a Trust management

Different authors have provided a number of solutions to secure Directed Diffusion against some of the attacks mentioned above. VijayRaman Kumar et al. [9] have proposed a solution to secure Directed Diffusion using a key management protocol called LEAP (Localized Encryption and Authentication Protocol). Different keys have been used to provide various levels of security in this approach. Each node shares a pair-wise key with all its neighbours for secure communication between them. All the nodes in the sensor network share a Global key with the base station which is used by the base station to encrypt the interest message and by other nodes to decrypt the messages from the base station. Each node has also a unique individual key which is used for secure communication between the node and the base station. The base station verifies the messages sent by this node using this key.

The approach taken by the authors can secure Directed Diffusion against cloning, flow suppression attacks but cannot effectively work against selective forwarding attack as the adversary can drop an encrypted packet at its will.

Another solution provided by Nashwa El-Bendary et al. [10] uses µTESLA (micro, Timed, Efficient, Streaming, Loss-tolerant Authentication) algorithm to authenticate acknowledgement messages sent from sink to source for confirming delivery of datamessages.
The main phases of the protocol are:

- Secret-key setup and broadcasting phase
- Interest propagation phase
- Path establishment and reinforcement phase
- Data-event delivery and authenticated acknowledgement delivery phase
- Disclosed key and buffered ACK-packets authentication phase

4.1.1 Threat Model and Assumptions

Directed diffusion is susceptible to selective forwarding, sinkhole, Sybil, wormhole and HELLO flood attacks. In our threat model, we consider the selective forwarding attack and two new types of attacks- Good mouthing attack and bad mouthing attack which are the threats to the trust mechanism itself.

- **Selective forwarding Attack:** An adversary can place malicious nodes at arbitrary places in the network. These nodes arbitrarily drop some of the packets routed through them. This leads to degradation of network performance.
- **Bad mouthing Attack :** In bad mouthing attack, the attacker propagates negative information about legitimate nodes in the network and tries to convince other nodes that a particular node is malicious.
- **Good mouthing Attack :** Good mouthing attack is carried out by a group of malicious nodes in which they propagate good reputation about malicious nodes in the network and try to convince other nodes that they are legitimate nodes.

4.1.2 Modeling reputation based Trust

We use the concept of trust to enforce security to the sensor nodes. Distributed nature of sensor networks only allow to have imperfect knowledge about others. Thus, every node can have an opinion about another node but it is difficult to be certain whether that node is malicious or not. This opinion is derived from belief (how much a node is trusted) , disbelief (how much a node is suspicious) and uncertainty where both belief and disbelief are lacking. This is expressed mathematically as [10]:

$$b+d+u=1$$

Here, b, d, u represents belief, disbelief and uncertainty respectively. A node considers observations from direct interactions as well as recommendation from other nodes (whom it trusts) to evaluate reputation of a node.

4.1.3 First-hand information gathering

First-hand information is gathered from a node's direct interaction with another node. Firsthand information is periodically propagated to the neighbours so that they can use it as second-hand information. Each node calculates its trust based on the observations as well as recommendations using Dempster Shafer Belief theory [10]. Here binomial opinions are collected that is a node is either rational or malicious. Thus beta distribution, Beta(α,β) is used here in which α and β are continuously updated as observations are made. Initially, all the nodes has Beta(1,1) for all its neighbours. When a new observation is made, if a node successfully forwards a packet it has received, its α is updated; otherwise, β is updated as weighted (ω) average of old and new observations according to the following equations:

$$\alpha ij\ new = \omega * \alpha ij\ (ol\ ) + 1 - \omega * \mu$$
$$\beta ij\ (ne\ ) = \omega * \beta ij\ old + 1 - \omega * \mu$$

When a node j receives a packet from node i and successfully forwards it to another node k; then node j sends an acknowledgement (ACK) packet to node i. If node j does not forward the packet; then no ACK is sent to node i. Node i updates Beta(α,β) of node j according to following equations with µ=2 in case of receipt of ACK and timeout respectively. High (α+β) implies that there is more evidence as the nodes frequently communicated and therefore, uncertainty (u) is low. Also, when α=β; it means that there is equal evidence for both success and failure of the transaction. Thus though the nodes communicated frequently, they did not behave consistently; hence

uncertainty (u) is very high. Thus, uncertainty (u) in predicting the nature of node$_i$ by node$_i$ is defined as:

$$u_{ij} = \frac{12 * \alpha_{ij} * \beta_{ij}}{(\alpha_{ij} + \beta_{ij})^2 * (1 + \alpha_{ij} + \beta_{ij})}$$

Therefore, total certainty=(1-*uij*). Total certainty is composed of both belief and disbelief. Hence, according to the proportion of supporting evidence available, belief and disbelief for nodej at nodei is calculated as [10]:

$$b_{ij} = \frac{\alpha_{ij}}{\alpha_{ij} + \beta_{ij}}(1 - u_{ij})$$

$$d_{ij} = \frac{\beta_{ij}}{\alpha_{ij} + \beta_{ij}}(1 - u_{ij})$$

4.1.4 Second-hand Information Integration

Second-hand information is the information a node gets from the first hand information received from other nodes. We consider second-hand information to estimate the behaviour of the neighbours as this lead to faster convergence and detection of malicious nodes. Here, nodek gathers opinion (in b,d,u) about nodej from each of its neighbours. After it receives all the recommendations from its neighbours about a node, it integrates the recommendation with the direct observation and finally decides how much that node can be trusted. Node k receiving (bj i, dj i uj i) from node i about node j, calculates [10]

$$b_j^{k:i} = b_i^k * b_j^i$$

$$d_j^{k:i} = b_i^k * d_j^i$$

$$u_j^{k:i} = b_i^k * u_j^i + d_i^k + u_i^k$$

where bj k:I , dj k:i , uj k:i denotes the belief, disbelief and uncertainty of nodek on nodej after getting recommendation from nodei. Each node broadcasts their first hand information to their neighbours periodically.

4.1.5 Combining direct and indirect observation

On receiving the periodic broadcast messages, nodek combines the recommendation about nodej from multiple nodes i as [11]

$$b_j^{k(indirect)} = \frac{\sum_{i=1}^{n} b_j^{k:i}}{n}$$

$$d_j^{k(indirect)} = \frac{\sum_{i=1}^{n} d_j^{k:i}}{n}$$

$$u_j^{k(indirect)} = \frac{\sum_{i=1}^{n} u_j^{k:i}}{n}$$

where n is the number of nodes from which k gets recommendation about j. Node k combines its direct and indirect opinion about node j as [11]

$$b_j^{k(final)} = \phi_1 b_j^k + \phi_2 b_j^{k(indirect)}$$

$$d_j^{k(final)} = \phi_1 d_j^k + \phi_2 d_j^{k(indirect)}$$

$$u_j^{k(final)} = 1 - b_j^{k(final)} - d_j^{k(final)}$$

where, $\phi_1 = \frac{\gamma * u_j^{k(indirect)}}{(1-\gamma)u_j^k + \gamma * u_j^{k(indirect)} - 0.5 u_j^k u_j^{k(indirect)}}$

$$\phi_2 = \frac{(1-\gamma) * u_j^k}{(1-\gamma)u_j^k + \gamma * u_j^{k(indirect)} - 0.5 u_j^k u_j^{k(indirect)}}$$

Here, γ is the weight given to direct observation. We assume that a node gives more weight to its own observation than the recommendation it receives from its neighbours. Hence, it is taken to be γ=0.6

4.1.6 Evaluation of trust and detection of malicious node

After combining the direct observation and recommendations received from others, a node decides whether another node can be trusted or not. Trust is calculated by nodek for nodej as [11]

$$T_{kj} = b_j^{k(final)} + 0.5\, u_j^{k(final)}$$

If Tkj is less than a predefined threshold of trust, nodek adds nodej to its list of malicious nodes. When a malicious node is detected, it broadcasts its direct observation about the node to its neighbours.

Effect of selective forwarding on different phases of Directed Diffusion and how the trust model detects the malicious nodes and avoids them

Interest propagation- The sink periodically broadcasts an interest message to each of its neighbours. If the receiving nodes are not the source, they broadcast the message to their neighbours till the source of the data is reached.

- **Effect of malicious node-** The malicious nodes either forward the interest message or drop it. If the interest message is dropped by some malicious node, still it will reach the source by some other path as the message is broadcast.
- **Trust model-** Due to the robust nature of interest propagation in Directed Diffusion, there is no need for the ACKs to be sent to the sender on successful forwarding of interest messages.

Exploratory data propagation- The source on receiving interest message from its neighbours sends data message to each of its neighbours from which it had received interest message. This is followed till the data message reaches the sink.

- **Effect of malicious node-** The malicious nodes can randomly forward some data message and drop some.
- **Trust model-** For each data message received and successful forwarding, the upstream node sends an ACK to the downstream node. Based on receipt of ACK or in case of timeout, the downstream node updates its opinion about the upstream node and subsequently calculates its trust on the upstream node.

Reinforcement- On receiving the exploratory data messages, the sink sends reinforcement message to any one of the downstream nodes. The node receiving the reinforcement message forwards the reinforcement message to one of its downstream nodes till it reaches the source.

- **Effect of malicious node-** A malicious node which has been reinforced may drop the reinforcement message. As a result, the reinforcement message will not reach the source and it will not be able to generate the required data packets.
- **Trust model-** The sink reinforces the downstream node with the highest trust (T). The reinforced node further reinforces a downstream node with the highest T. This is followed till the reinforcement message is received by the source. Thus, in our trust model, always the most trusted path is selected. In the beginning, when there is no information about the nodes and the trust on all the nodes is same, any of the nodes can be reinforced but if it is malicious, then in due time, it will be detected as its trust level will go down below the threshold.

Data propagation- The source sends data through the reinforced path.

- **Effect of malicious node-** A malicious node present in the data path may drop some packets randomly, thus, disrupting the flow of data.
- **Trust model-** Since, we have already reinforced the path with the highest trust, the probability of selective forwarding attack is less. In this phase also, the downstream node updates its opinion of the upstream node based on receiving ACK as well as the recommendations from other nodes and subsequently updates the trust. But, a node's behaviour may change at runtime and hence, it may drop some packets. In this case, data will not reach the upstream nodes and finally to the sink. So, when data is not received, a node can negatively reinforce a previously reinforced path and reinforce a other path with high trust. The motive here is to choose a highly trusted path so that the chances of selective forwarding are significantly lower than in normal case.

Bad mouthing attack and **Good mouthing attack** are threats to any trust model. As we adopt a policy of exchanging observations among the neighbors, a malicious node may try to propagate bad reputation about normal nodes (bad mouthing attack) or propagate good reputation about another malicious node (good mouthing attack).

V.EXPERIMENTAL RESULT

An OTcl script written by a user is interpreted by NS. While OTcl script is being interpreted, NS creates two main analysis reports simultaneously. One of them is NAM (Network Animator) object that shows the visual animation of the simulation. The other is the trace object that consists of the behavior of all objects in the simulation. Both of them are created as a file by NS. Former is .nam file used by NAM software that comes along with NS. Latter is a ".tr" file that includes all simulation traces in the text format.

Simulation Parameters

The simulation is carried out in NS-2. Various simulation parameters are listed in table 1.

Sr.No.	Parameter	Value
1	Simulator	NS-2 (Version 2.35)
2	Channel Type	Channel / Wireless Channel
3	Radio Propagation Model	Propagation / Two ray ground wave
4	Network Interface Type	Phy / WirelessPhy
5	MAC Type	Mac/802.11
6	Link Layer Type	LL
7	Antenna	Antenna / Omni Antenna
8	Area	800 * 800
9	Simulation Time	1800 s
10	No. of Nodes	30

Table 1 : Simulation parameters

First an example is cited for better understanding of the protocol. Then the results are discussed. As each node forwards packets to an upstream node, it waits for an ACK from the upstream node. The upstream node on forwarding the received packet to its subsequent upstream node sends an ACK packet to the downstream node. On receiving ACK from the upstream node or in case of timeout, the observing node updates its trust on the upstream node according to the algorithm described.

Here, node 27 which is the source generates a packet and forwards it to its neighbour node 2 (Figure 6) and waits for an ACK from node 2 on successful forwarding of the packet by node 2. Node 2 sends an ACK to node 27 on successful forwarding of the packet (Figure 7) and it does not send any ACK to the sender when it does not forward the packet (Figure 8). Based on receiving ACK from node 2 or in case of timeout, node 27 evaluates its trust on node 2.

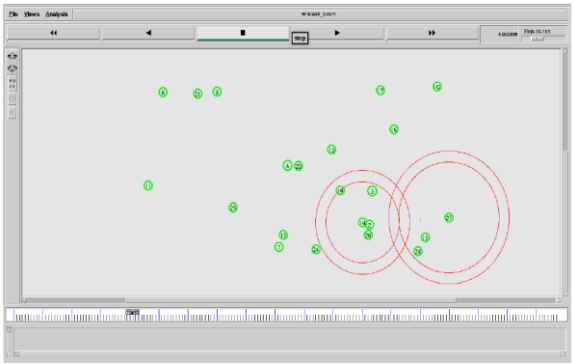

Figure 6: Packet forwarding by the source to its neighbour

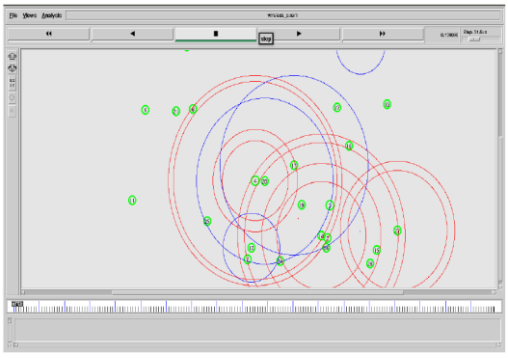

Figure 7: ACK sent by node 2 to node 27 on forwarding the received packet

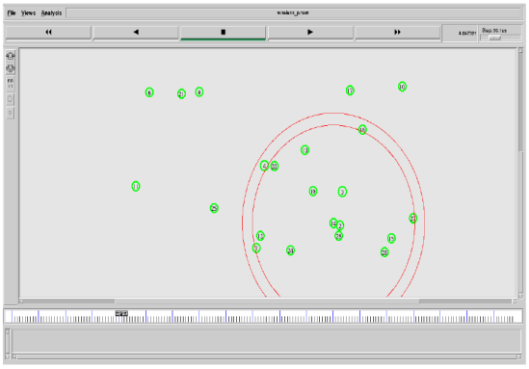

Figure 8: Packet dropped by node 2 and no ACK sent to node 27

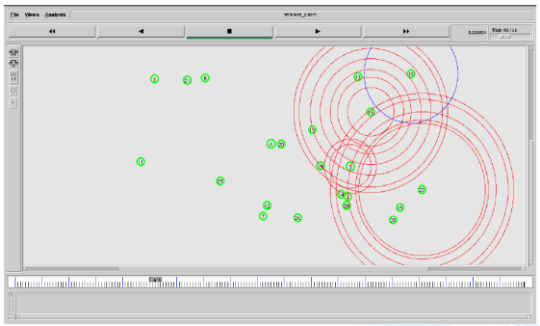

Figure 9: Alternate neighbour selected for packet forwarding

Since node 27 has detected node 2 as malicious 2. , therefore it stops transactions with it and forwards its packets to node 16. The following graph shows the result of how the proposed scheme works better than without scheme.

The graph shows the comparative performance of the Directed Diffusion protocol with and without the proposed scheme. The performance metric considered is Packet Delivery Ratio. The packet delivery ratio is defined as the ratio of number of data packets received at the destination over the number of data packets sent by the source.

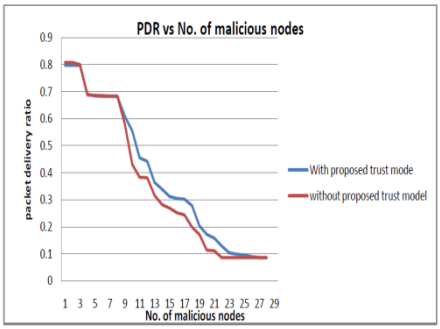

Figure 10: Variation of packet delivery ratio with no. of malicious nodes

Here, we can see that the packet delivery ratio of the Directed Diffusion protocol with the proposed scheme is greater than without the scheme. Initially, the performance of the Directed Diffusion protocol with and without the scheme is similar till there are about 10% malicious nodes in the network. But, from thereon, Directed Diffusion with the proposed scheme performs considerably better till there are about 89% malicious nodes in the network. This is because, after detection of malicious nodes, the legitimate nodes refrain from any transactions with them. Hence, the packet delivery ratio of the proposed scheme is greater. However, when all the nodes in the network become malicious then the performance of Directed Diffusion with or without the scheme is similar.

The next graph shows how the delivery ratio increases with time after a certain point. Initially, all the nodes are equally trustworthy. The experiment is carried out considering node 8 launches selective forwarding attack. As a result, the packet delivery ratio decreases initially. But gradually, with increase in time the malicious nodes are detected and as a result, the packet delivery ratio increases.

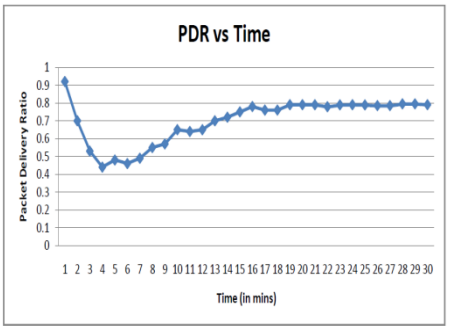

Figure 11: Packet delivery ratio as a function of time with the proposed trust model

VI.CONCLUSION

In this work, we have proposed a trust and reputation based secure Directed Diffusion routing protocol for wireless sensor network. However this trust model so developed can be readily extended for other routing protocols in order to deal with the problems of selective forwarding, good mouthing and bad mouthing. Simulation is carried out in NS-2 to find the performance of the network with and without the proposed scheme. The simulation results show that our trust model increases the packet delivery ratio of the network against selective forwarding attack. It is assumed that malicious nodes send acknowledgement to the sender when it forwards the packets received. We have not considered the case when it sends acknowledgement even when it drops the received packets. We plan to address this issue in future. Also, we would consider starting our algorithm when a significant drop in performance of the network is observed. This would reduce the overhead caused by the acknowledgement scheme used in our work. We plan to extend this model for the other types of attacks which we have not included in our present work. Also we would like to extend this work for mobile sensor networks.

VII. REFERENCES

[1]. S. Basagni, K. Herrin, D. Bruschi, E. Rosti, \Secure Pebblenets," in Proc. 2nd ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 156{163, 2001.

[2]. M. Bellare, S. Miner, \A Forward-Secure Digital Signature Scheme," in Proc. Annual Int. Cryptology Conference (CRYPTO), Lecture Notes in Computer Sci- ence, vol. 1666, pp. 431-448, 1999.

[3]. W. Du J. Deng, Y. S. Han, P. K. Varshney, \A Witness-Based Approach for Data Fusion Assurance

in Wireless Sensor Networks" in Proc. IEEE GLOBECOM, pp. 1435{1439, 2003.

[4]. J. Deng, R. Han, S. Mishra, \Security Support for in-Network Processing in Wireless Sensor Networks," in Proc. 1st ACM workshop on Security of Ad Hoc and Sensor Networks, pp. 83{93, 003}.

[5]. D. Ma, G. Tsudik, \DISH: Distributed Self-Healing (in Unattended Sensor Networks)," Cryptography ePrint Archive, Report 2008/158, pp. 47{62, 2008.

[6]. R. Anderson, \Two Remarks on Public-Key Cryptology,invited lecture", in Proc. 4th ACM Conf. on Computer and Communications Security, 1997.

[7]. D. Ma, C. Soriente, G. Tsudik, \New Adversary and New Threats: Security in Unattended Sensor Networks," IEEE Networks, vol. 23, no. 2, pp. 43{48, 2009.

[8]. X. Chen, K. Makki, K. Yen, N. Pissinou, \Sensor Network Security: a Survey," IEEE Communications Surveys and Tutorials, vol. 11, no. 2, pp. 52 {73, 2009.

[9]. M. Sirivianos, D. Westho_, F. Armknecht, J. Girao, \Non-Manipulable Aggregator Node Election Protocols for Wireless Sensor Networks," in Proc. Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 1{10, 2007.

[10]. R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, G. Tsudik, \Data Security in Unattended Wireless Sensor Networks," IEEE Transactions on Comput- ers, vol. 58, no. 11, pp. 1500{1511, 2009.

[11]. A. S. S. Mateus, C. B. Margi, M. A. S. Jr., C. C. F. P. Geovandro, B. T. de Oliveira, \Implementation of Data Survival in Unattended Wireless Sensor Networks Using Cryptography," in Proc. IEEE Conf. on Local Computer Net- works (LCN), pp. 961{967, 2010.

