

ESTIMATION BASED FRAMEWORK FOR IDENTIFYING MALICIOUS DATA INJECTIONS IN WIRELESS SENSOR NETWORKS

Sailaja Gokavarapu, M.Tech¹ and Md. Abdul Azeem, Associate Professor²

^{1,2}Department of CSE, MVSR Engineering College, Nadergul, Hyderabad

Abstract— Wireless Sensor Networks are widely advocated to monitor environmental parameters, structural integrity of the built environment and use of urban spaces, services and utilities. However, embedded sensors are vulnerable to compromise by external actors through malware but also through their wireless and physical interfaces. Compromised sensors can be made to report false measurements with the aim to produce inappropriate and potentially dangerous responses. Such malicious data injections can be particularly difficult to detect if multiple sensors have been compromised as they could emulate plausible sensor behaviour such as failures or detection of events where none occur. A novel algorithm is proposed to identify malicious data injections and build measurement estimates that are resistant to several compromised sensors even when they collude in the attack. A methodology is also proposed to apply this algorithm in different application contexts and evaluate its results. The algorithm consists of three phases viz., Estimation, similarity check and characterization. In similarity check, there are two tests that capture the characteristics of most event detection criteria. The magnitude test verifies that reported measurements are close in magnitude to their estimates. The shape test verifies that the estimate and reported signal have a similar shape. Base work only concentrated on the detection of malicious data injections, the entire process is centralized and is being carried out at the base station. The base work is enhanced to distributed architecture. As it is an in-network process, the process of detection of malicious injections is evenly distributed in the network. In order to avoid transmission of malicious data through the network nodes and to curtail the energy wastage in network, the detection is done at the cluster head level itself by maintaining the accuracy using the LEACH characteristic.

Keywords—security management; adhoc and sensor networks; statistical methods; Malicious data injections; measurements analysis

I. INTRODUCTION

wireless Sensor Networks are spatially distributed autonomous sensors to monitor physical or environmental condition, such as temperature, sound, pressure, etc and to co-operatively pass their data through the network to a main location. They are often used to detect events occurring in the physical space across different applications such as military surveillance[1], health[2], and environment (e.g. Volcano)[3] monitoring etc. Although these applications have different tasks, they all collect sensor measurements and interpret them to identify events, i.e., particular conditions of interest followed by a remedial response. Such response may have significant consequences and cost. Therefore, the measurements leading to the event detection, become a critical resource to secure. When the measurements are somehow replaced or modified by an attacker, we deal with malicious data injections. The attacker may make use of the injected data to elicit an event response, such as evacuation in case of fire, when no event has occurred, or mask the occurrence of a true event, such as the trigger for an intrusion alarm. Different means for obtaining control over the measurements are possible.

A wireless sensor network (WSN) consists of sensor nodes capable of collecting information from the environment and communicating with each other via wireless transceivers. The collected data will be delivered to one or more sinks, generally via multi-hop communication. The sensor nodes are typically expected to operate with batteries and are often deployed to not-easily-

accessible or hostile environment, sometimes in large quantities. It can be difficult or impossible to replace the batteries of the sensor nodes. On the other hand, the sink is typically rich in energy. Since the sensor energy is the most precious resource in the WSN, efficient utilization of the energy to prolong the network lifetime has been the focus of much of the research on the WSN.

In wireless sensor network data gathering and routing are challenging tasks due to their dynamic and unique properties. Many routing protocols are developed, but among those protocols cluster based routing protocols are energy efficient, scalable and prolong the network lifetime.

II. PROBLEM STATEMENT, PROPOSED SOLUTION, MOTIVATION

A. Problem statement

Here it is considered directly the scenario where an attacker gains full control of one or more sensors and can run arbitrary malware on them to fabricate new measurements and report them in place of the observed ones.

This task consists of detecting the incongruities between the observed and the reported measurements. To detect malicious data injections, an algorithm is proposed that characterizes the relationships between sensors' reported values arising from the spatial correlations present in the physical phenomenon.

B. Proposed solution

We propose a novel algorithm to identify malicious data injections and build measurement estimates that are resistant to several compromised sensors even when they collude in the attack. We introduce novel ways of aggregating measurements that are aimed at discarding malicious contributions under attack and minimize the false positives under genuine circumstances as well. We also propose a novel more general methodology to apply our algorithm in different application settings.

We describe the three phases our algorithm those are Estimation, similarity check and characterization. In similarity check we propose two tests that capture the characteristics of most event detection criteria. The magnitude test verifies that reported measurements are close in magnitude to their estimates. The shape test verifies that the estimate and reported signal have a similar shape.

C. Motivation

Measurements of two sensors are related and in particular spatially correlated. Measurements are correlated under genuine circumstances, compromised measurements disrupt such correlations. Each sensor can exploit correlations to produce an estimate for the measurements of other sensors. Since the estimates are directly calculated from the raw measurements, it does not introduce additional variables. The estimates can then be aggregated with a collusion-resistant operator that produces a final reliable estimate to be compared with the reported measurement.

D. Objective

Reduction of computation and communication costs incurred in detection of malicious data injections by using measurement estimates. And to develop a general methodology to flexibly tailor the technique to WSN applications that detect different kinds of events. Besides these the other objective is to curtail energy loss and to enhance the life time of the network.

III. LITERATURE SURVEY

3.1 Related Work

There are different techniques proposed to detect malicious data injections namely Software attestation techniques, Majority voting and Trust management framework etc. Each of them is discussed in the foregoing sections. In addition, the hierarchical routing has also been discussed.

3.1.1 Software Attestation Techniques. In the presence of malicious data injections, there are few observable properties that can help detection. One of them is the loss of integrity of the sensor eg.

that it is running malicious software. For such a scenario, software attestation techniques [4]-[6] have been proposed. But require further evaluation in concrete deployments.

However, that injections through environment manipulation (The attacker manipulates the environment by using for instance a lighter to trigger a fire alarm) cannot be detected through attestation since software is still genuine.

3.1.2 Majority Voting. An approach based on aggregation of individual sensor's information [7]-[10], where each sensor votes for neighbor's maliciousness and votes are aggregated by majority. It introduces an additional variable – the vote. Detecting such attacks incurs additional computation and communication costs.

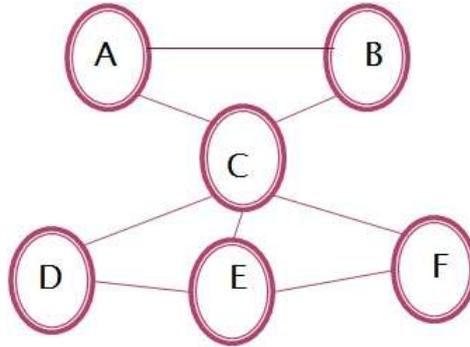


Figure 1. Example WSN Topology

In the above Fig 1, Nodes represent sensors, edges indicate a neighbourhood relationship. Consider first nodes A, B and C to be compromised. In this case A is free to inject arbitrary malicious data if B and C collude to not report on it and act genuinely to avoid reports from D, E, F. If estimates were available we would notice that the measurements of B and C are consistent with D, E, F, but are not consistent with those of A.

Alternatively consider nodes D, E, F, to be compromised. Here nodes D and E can inject any kind of measurements, although C may report on them. Indeed, node F can avoid reporting on them and report on C instead. Then with simple majority voting approach node C would appear as the compromised node. Majority voting approach will always fail when more than 50% of sensors are compromised.

3.1.3 Trust Management Frame Work. A Sensor's behaviour is mapped to a trust value by all its neighbours, and then the sensor's trustworthiness is obtained by averaging the trust values[11]-[14]. The main draw back of these techniques is that they introduce an additional variable trust value about which an attacker can lie with or without lying about the measurements at the same time.

3.1.4 Hierarchical Routing. LEACH (Low Energy Adaptive Clustering Hierarchy) is the first network protocol that uses hierarchical routing for wireless sensor networks where all the nodes in a network organize themselves into local clusters, with one node acting as the cluster-head[15][16]. All non-cluster-head nodes transmit their data to the cluster-head that transmits data to the remote base station.

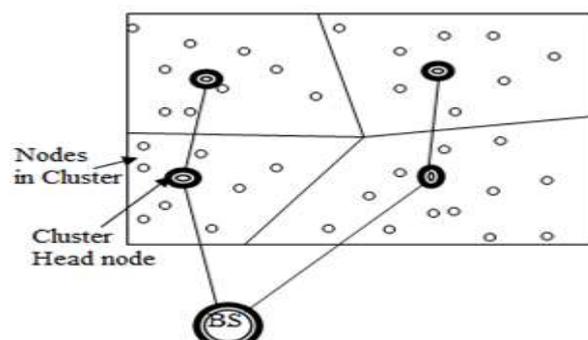


Figure 2. LEACH Network Design

In Fig 2, all non-cluster head nodes transmit their data to the cluster-head, while cluster head node receives data from all the cluster members, perform signal processing functions on the data (eg. data aggregation) and transmit data to the remote base station. The job of cluster head rotates here, which results in balancing the energy expense, saves the node energy and prolongs the life time of the network.

3.2 Summary

Majority voting approach is based on aggregation of individual sensors information. Similarly, Trust Management frameworks aggregate individual beliefs about a sensor's behaviour. Both of these techniques introduce an additional variable – the vote, or trust value – about which an attacker can lie with or without lying about the measurements at the same time.

Detection of such attacks incurs additional computation and communication costs. However, that injections through environment manipulation cannot be detected through attestation since software is still genuine.

3.3 Gap in the Existing Research and Need of Today/Scope for Improvement

Majority voting approach will always fail when more than 50% sensors are compromised. There is a need to show tolerance against more no.of compromised nodes and to reduce computation and communication costs. Algorithms used in prior studies [17] [18] cannot be systematically tailored to different deployments and different applications. Robustness is required.

IV. PROPOSED FRAME WORK FOR IDENTIFYING MALICIOUS DATA INJECTIONS IN WSNS

4.1 Methodology

The Estimation-based framework [19], which iteratively extracts and aggregates measurements estimates, is at the core of detection mechanism. Estimates are iteratively computed on new measurements and a similarity check compares them as shown in Fig 3. When similarity check fails, we run a characterization step – an extensive analysis that identifies the likely compromised sensors.

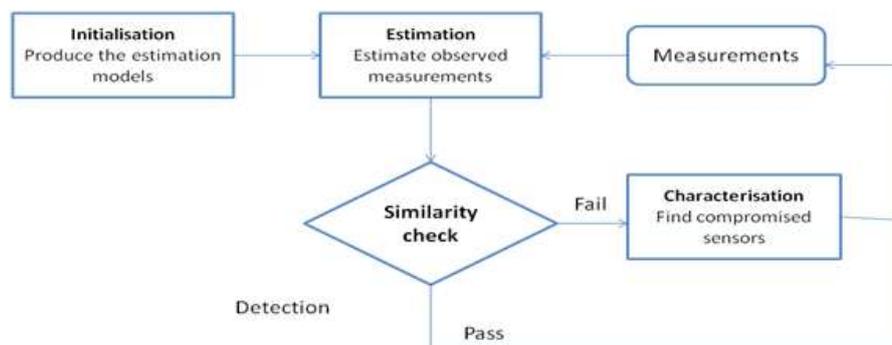


Figure 3. Outline of the framework

4.1.1 Initialization. Sensor Measurements are aligned with inter sensor delay.

4.1.2 Estimation. Compromised sensors could collude to bias the estimate and make it more consistent with the reported measurement. To avoid this problem, separate pairwise estimates are calculated with each neighbour. In a second step estimates are aggregated by an operator that is resistant to compromised estimates.

4.1.2.1. Pairwise Estimation. The measurements of two sensors are related, and in particular spatially correlated, because the sensed physical phenomena affect and propagate across the environment in which the sensors are placed. The relationship could be characterized in a mathematically precise way, given by the laws of physical phenomenon and its propagation. If node

'k' has n neighbours, n number of estimates will be obtained using linear regression. Each estimate carries its own weight, also known as prior weight.

4.1.2.2 Aggregated Estimation. Estimates are aggregated by an operator i.e resistant to compromised estimates. Two candidates to aggregate pairwise estimates are weighted mean and weighted median. Both take as input a set of estimates and their prior weights and return an aggregated value. The weighted mean can achieve a smaller error than those of single estimates. It is highly sensitive to compromise. In contrast weighted median is more resistant to compromise. It first sorts the values ascendingly, then arranges the weights with the same order, then picks the element at the half length. Its drawback is that by picking one among all estimates, the error cannot be reduced further.

Since there is a trade off between accuracy and compromised resistance, we propose to combine the two operators. First the weighted median operator is applied, then the weighted mean is calculated with new weights (posterior weights). Posterior weights are obtained as the prior weight times a function (Complementary Cumulative distribution function of the estimated error).

4.1.3 Similarity Check. Reported measurement and estimate of the observed value O_I are compared using Similarity metric that must be consistent with the event detection criterion. So two signals that are similar according to the metric must also have similar effects on the event detection and vice-versa.

4.1.3.1 Magnitude Test. Verifies the reported measurements are close in magnitude to their estimates.

4.1.3.2 Shape Test. Verifies that the estimate and reported signal have a similar shape by using the deviations calculated from coefficient(Pearson correlation coefficient), which were obtained using pairwise estimates.

4.1.4 Characterization. Characterization step consists in removing the sensors with the highest deviation, one by one, and re computing the similarity check on the remaining sensors in the neighbourhood. Each time we remove a sensor, which we presume compromised, the genuine sensors gain in consistency with their estimate whereas colluding sensors lose the benefits of the removed sensor's estimate. The procedure stops when all the remaining sensors pass the similarity check. And returns the compromised sensors as output.

V. IMPLEMENTATION OF ESTIMATION BASED FRAMEWORK

5.1 Protocol Description

5.1.1 Hierarchical Routing Protocol (LEACH). Divise the network into groups that communicate through their Cluster Heads (CH). Low Energy Adaptive Clustering Hierarchy(LEACH) is a hierarchical routing protocol. Some nodes of LEACH network act as Cluster Heads. The job of the cluster-head is to collect data from their surrounding nodes and pass it on to the base station. LEACH is dynamic because the job of cluster-head rotates. The LEACH network has two phases: the set-up phase and the steady-state. In the Set-Up Phase Cluster Heads are chosen and in the steady state the cluster head is maintained when data transmitted between nodes. The operation of LEACH is illustrated in Fig 4.

A,B,C,D are the cluster members. E is the Cluster Head (CH) and BS is the Base Station. The Cluster Head collects information from multiple nodes called cluster members and passes it to Base station.

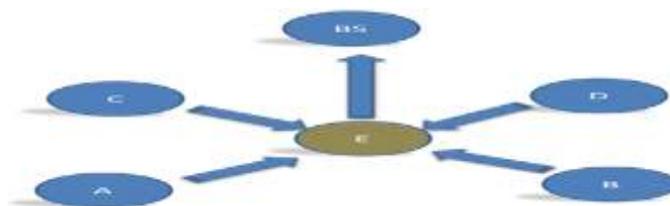


Figure 4. Communication through CH

5.1.2 Ad hoc on Demand Distance Vector (AODV). Ad hoc on demand distance vector (AODV) routing protocol creates routes on-demand. In AODV, a route is created only when requested by a network connection and information regarding this route is stored only in the routing tables of those nodes that are present in the path of the route. AODV protocol uses a single route reply message.

5.1.3 Reverse AODV. The modified AODV (R-AODV) protocol [20] discovers routes on demand using a reverse route discovery procedure. During route discovery procedure source and destination nodes play some role from the point of sending control messages. Thus after receiving RREQ message the destination node floods reverse request (R – RREQ), to find source node. When the source node receives an R-RREQ message, data packet transmission is started immediately. Here route reply message is multicast to its neighbors resulting in redundant route reply messages.

5.2 Project Overview

The Framework is implemented using the tool NS2. There are total 1500 nodes in the network, which have been divided into 300 groups. Nodes communicate with base station by using LEACH protocol. Each group consists of 5 nodes and from each group one node can act as Cluster head (CH). In this cluster based approach the sensors do not need to communicate with the base station directly. Instead CHs are responsible to send the data collected within the BS. And the job of CH rotates here. Cluster head selection process will be done at 0.6 sec. Route discovery process is done at 0.8 sec by using RAODV protocol. And after every 15 Sec Cluster heads change.

Spatial correlations exist amongst sensors of each group. Sensors sense the values and transmit to base station through cluster heads. Here the attacks may compromise the measurements even before they are transmitted. By using the proposed methodology we can detect such type of malicious injections and remove the compromised nodes from data transmission process.

5.2.1 Cluster Head Selection Process. For the first time, all the member of the cluster are asked to generate a random number. And if the random number is less than a pre determined threshold that particular node becomes a cluster head for the current round. Next time onwards the energy levels of the sensors are compared, the node with highest energy can act as a CH. In this process if more than one node have same energy (highest value), then they have to generate a random number and the same process is to be followed as first round. Once a node is acted as a cluster head, it cannot become the cluster head next time onwards.

5.2.2 Assumptions and Dependencies. Initially all the sensor nodes have the same energy. Based on the data transmission the energy loss will vary in each sensor nodes. Each node can able to communicate to the base station directly. Each sensor can adopt its coverage area based on the situation (CH and CM operations).

5.3 Algorithms of Different Phases

All the algorithms are implemented using OTCL and C++ and Window size is taken as 5.

5.3.1 Pairwise Estimation. Separate Pairwise Estimates are calculated with each neighbor.

Table 1. Symbols used in pairwise estimation algorithm

S	Generic sensors deployment in WSN
O_i	Refers to a sample of the random variable, contains measurement of sensor i.
O_j	Refers to a sample of the random variable, contains measurement of sensor j.
$N(i)$	Neighbor set of Sensor i
O_{ij}	Estimate of O_i based on O_j

Algorithm 1 Pairwise Estimations calculation

```

I.   INPUT     $O_i, i \in S$ 
II.  OUTPUT    $O_{ij}$ 
III.  for all  $i \in S$  do
IV.   for all  $j \in N(i)$  do
V.     Calculate and store pairwise estimates  $O_{ij}$  using linear regression.
VI.   end for
VII.  end for
    
```

NOTE: Estimate of Sensor A based on B using Linear regression = $a_{ij} B + b_{ij}$

$$a_{ij} = \text{Cov}(O_i, O_j) / \text{Var}(O_i)$$

$$b_{ij} = E[O_i] - a_{ij} E[O_j]$$

5.3.2 Aggregation of Pairwise Estimates. Pairwise estimates are aggregated by an operator that is resistant to compromised estimates.

Table 2. Symbols used in aggregation of pairwise estimations algorithm

$W_{iN(i)}$	Indicate prior weights (to weigh neighbors contribution)
$O_{iN(i)}$	Estimates for i 's observed measurement from its neighbors
O_i	Aggregated estimate of sensor i

Algorithm 2 Aggregation of pairwise estimations algorithm

```

I.   INPUT  $W_{iN(i)}, O_{iN(i)}$ 
II.  OUTPUT  $O_i$ 
III. Calculate weighted median
IV.  for all  $j \in N(i)$  do
V.   Calculate the posterior weights
VI.  end for
VII. Calculate  $O_i$  by taking the weighted mean of pairwise estimates along with posterior weights.
    
```

NOTE: Posterior Weight calculation:

- i. Calculate the weighted median of the inputs
- ii. Apply weighted median along with each of estimate to a function P.
- iii. P is equal to $1 - \text{erf}(\text{abs}(\text{weighted median} - \text{estimate}) / \text{residual SD})$
- iv. Multiply the above result with prior weight
- v. Store the results of previous step in an array $W_-()$
- vi. Each element of $W_-()$ divided by the sum of elements, gives the posterior weight corresponding to that element

Function P penalises values distant from the weighted median. Such function is the complementary cumulative distribution function of the estimation error and erf() is the error function.

5.3.3 Similarity Check. Reported measurement S_i and estimate of the observed measurement O_I are compared using Similarity metric.

5.3.3.1 Magnitude Test. Verifies the reported measurements are close in magnitude to their estimates.

To Build Magnitude Test:

$$M_i = (O_I - S_i) \text{ [} S_i \text{ is the reported measurement]}$$

$$\text{The error } \epsilon_i = (O_I - O_i)$$

$$O_I = \text{true value} + \text{weighted mean of residuals}$$

$$\text{So Magnitude deviation} = M_i / \text{std}(\epsilon_i)$$

5.3.3.2 Shape test. Shape test verifies that the estimated and reported signal have a similar shape.

Table 3. Symbols used in shape test algorithm

S	Generic sensors deployment in WSN
O_i	Refers to a sample of the random variable, contains measurement of sensor i.
N(i)	Neighbor set of Sensor i
O_{ij}	Estimate of O_i based on O_j

Algorithm 3 Shape test algorithm

-
- I. **INPUT** $O_i, O_{ij} \quad i \in S, j \in N(i)$
 - II. **OUTPUT** $DR_i(O_i)$
 - III. **for all** $j \in N(i)$ **do**
 - IV. Calculate pearson correlation coefficients for a sensor i with the estimates given by j.
 - V. Calculate median from the coefficients.
 - VI. Store the medians in a list.
 - VII. Sort the median list.
 - VIII. Select the smallest element.
 - IX. Calculate the deviation percentage of difference between smallest element and coefficient.
 - X. Store the deviation percentages in a list.
 - XI. **end for**
 - XII. Deviation percentage corresponding to largest median is stored in a separate list(DR_i).
 - XIII. Deviation list (DR_i) is returned as output.
-

5.3.4 Characterization. Characterization step consists in removing the sensors with the highest deviation, one by one and recomputing the similarity check on the remaining sensors.

Table 4. Symbols used in Characterization algorithm

S	S is any particular group in WSN
$DR_i(O_i)$	Deviations obtained for sensor i during Shape test or Magnitude test
T	T is a predetermined threshold (Assumption), which varies with sensed values
K	K is the node with highest deviation in the group

Algorithm 4 Characterization algorithm

```

I.   INPUT    DRi (Oi)  □ i ∈ S
II.  OUTPUT  CompromisedSet
III.  CompromisedSet = { }
IV.  ResidualSet = S
V.   while (SimilarityCheck (DRi) fails) do
VI.      K ← Max(DRi)
VII.     if K. deviation > T then
VIII.        K is added to CompromisedSet
IX.        K is removed from ResidualSet
X.      for all j ∈ S do
XI.         Recompute DRj (Oj)
XII.     end for
XIII. end while
    
```

VI. RESULTS AND DISCUSSIONS

6.1 Simulation details

The estimation based framework for detecting malicious data injections has been implemented using the tool NS2. There are total 1500 nodes in the network, which have been divided into 300 groups, each consisting of 5 nodes. The sensors reported values are ranging from 1 to 9. At every second the sensors sense values. In every group the sensors, which are nearer to each other will sense closest value and vice versa.

Nodes communicate with base station through Cluster heads by using LEACH protocol. The proposed algorithm starts working after collecting five measurements of sensors (since 5 has been taken as window size) in both centralized and distributed architecture.

Table 5 shows the values of different parameters like mobility model, simulation time, topology, routing algorithm, no.of levels etc of simulation.

6.2 Parameters of Simulation

6.2.1 Life Time. Life time is the amount of time that a wireless sensor network would be fully operative that is the time at which the first network node runs out of energy to send a packet, because to lose a node means that the network could lose some functionalities.

Life time of network is more in distributed architecture.

6.2.2 Accuracy. Accuracy is in terms of identifying genuine and malicious data. It remains same in both centralized and distributed architecture.

6.2.3 Remaining Node Energy. It is the energy remaining in the sensors at the end of simulation. Remaining node is more in distributed architecture.

Table 5. Parameters of simulation

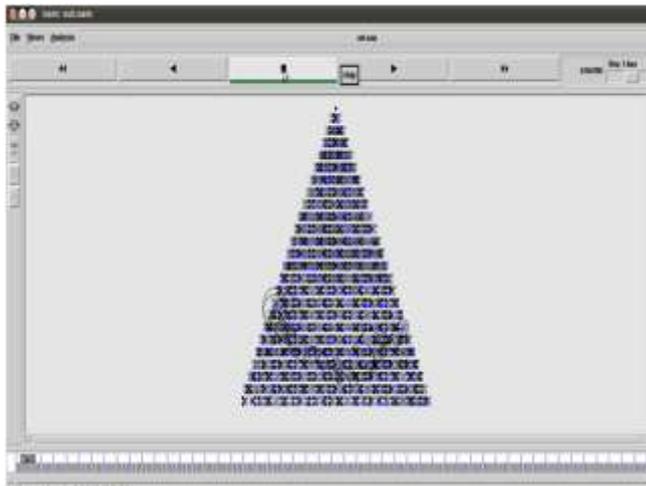
CRITERION	VALUE
TOOL	NS2
TOPOLOGY	Wireless Mesh
ROUTING ALGORITHM	LEACH RAODV (For Route Discovery)
PARAMETERS TO BE SIMULATED	Energy, Accuracy, Network life time
MOBILITY MODEL	Structured Model
SIMULATION TIME	35 Sec
NO. OF NODES	1500 Nodes plus 1 Base station

NO. OF CLUSTER HEADS	300
NO. OF LEVELS	24
CLUSTER HEADS SELECTION TIME	At time 0.6 Sec
CHANGING OF CLUSTER HEADS	At every 15 Sec
SENSORS SENSE THE DATA AND SEND TO BASE STATION	At every 1 Sec
MALICIOUS DATA INJECTIONS	15 Sec onwards
PROPOSED ALGORITHM STARTS WORK	6 Sec onwards

6.3 NAM File

NAM stands for Network Animator, this file is used for animating network actions.

6.3.1 Route Discovery Process. The nodes with blue color are cluster heads. Here the route discovery process is carried out by the base station using RAODV protocol as shown in Fig 5.



*Figure 5. CH Selection-At time 0.6Sec
Route Discovery-At time 0.8Sec*

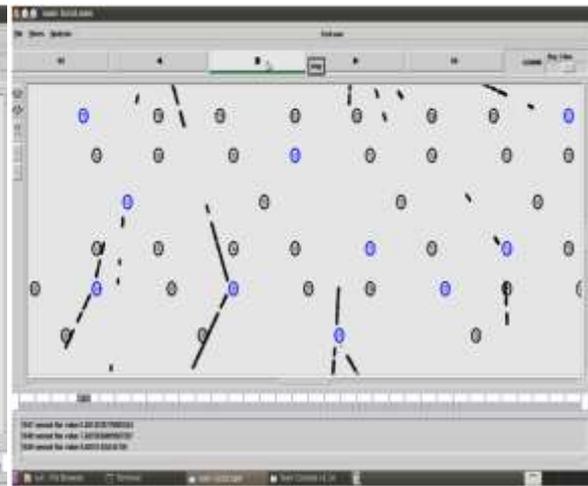


Figure 6. Sensed values passing through CHs in each second

6.3.2 Data transmission process through Cluster Heads. At every second, the sensed values of sensors pass through cluster heads to base station. The nodes with blue color are cluster heads. There are total 24 cluster levels. Cluster heads collect data from multiple nodes and pass to other cluster heads at higher level for sending the data to base station. Only few of cluster heads can communicate directly with the base station.

In Fig 6, Reported values of each node is printed at the bottom of the NAM window.

6.3.3 Transmission of Malicious data. The data flow with red color is the malicious data. These injections start at the time of 15 sec onwards. In Fig 7, the topmost node (With Node Id as Zero) acts as the base station. In the existing system, malicious data is also accepted by the base station as it cannot distinguish malicious and genuine data.



Figure 7. Malicious data approaching to Sink and accepted by Sink-At time 15 Sec onwards



Figure 8. Base Station Dropping the Malicious Data-At time 20Sec onwards

6.3.4 Base Station Drops the Malicious Data (Centralized Architecture). By using the 3 phases of the proposed algorithm, the base station is able to detect the malicious data injections (shown in red color) and drops them as shown in Fig 8. The process is done in centralized manner and is being carried out by the base station.

6.3.5 Dropping of Malicious Data at CH level (Distributed Architecture). In order to avoid transmission of malicious data through the network nodes and to enhance the life time of the network, the detection and dropping is done at the cluster head level itself as shown in the Fig 9.



Figure 9. CHs Dropping Malicious Data using distributed architecture

6.4 Graphs

XGraphs are used for analyzing output.

6.4.1 Reported Values vs Time Graphs. X-axis represents time varying from 1 to 35 Sec and Y-axis represents sensed values (reported) ranging from 1 to 9.

6.4.1.1 Reported values of sensors 541-550. The measurements of sensors under genuine circumstances are shown in Fig 10.

6.4.1.2 Reported values of sensors 611-620. The measurements of sensors under genuine circumstances are shown in Fig 11.

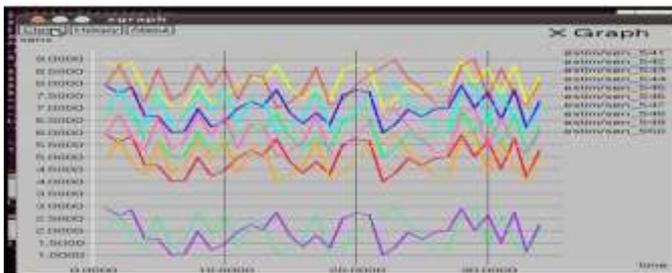


Figure 10. Reported values of 10 sensors(541-550)

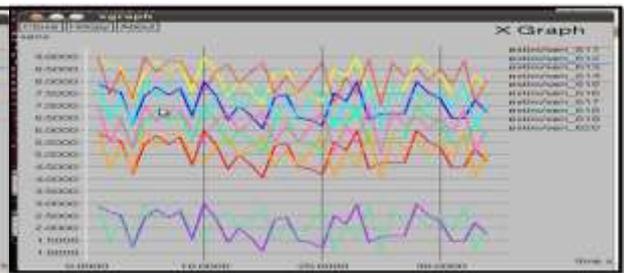


Figure 11. Reported values of 10 sensors (611 - 620)

6.4.1.3 Reported values of sensors 121-130. The measurements of sensors in presence of malicious injections are shown in Fig 12. Node 126 is compromised and correlations are disrupted. Values are shown in Table 6. Compromised sensor values are shown in BOLDFACE.

6.4.1.4 Reported values of sensors 201-210. Nodes 203 and 204 are compromised and correlations are disrupted. Fig 13 shows the presence of Malicious data injections. Reported values of sensors 201-210 are shown in Table 7. Compromised sensor values are shown in BOLDFACE.

6.4.1.5 Reported values of sensors 1-10. Nodes 3,4 and 5 are compromised and correlations are disrupted. Fig 14 shows the presence of Malicious data injections. Reported values of sensors 1-10 are shown in Table 8. Compromised sensor values are shown in BOLDFACE.

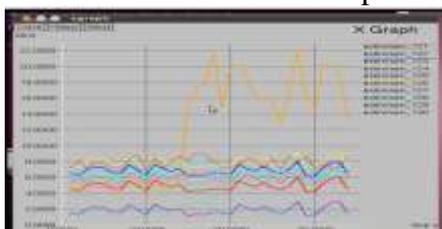


Figure 12. Reported values of 10 sensors(121-130)

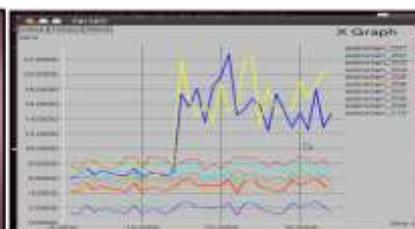


Figure 13. Reported values of 10 sensors(201-210)

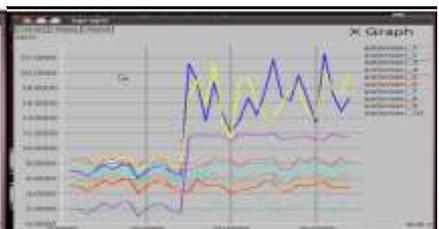


Figure 14. Reported values of 10 sensors (1-10)

Table 6 Reported values of sensors 121 – 130

121	122	123	124	125	126	127	128	129	130
5.62366	6.62273	7.62334	8.62294	2.61269	5.48832	6.48862	7.4882	8.48872	2.47804
5.12762	6.12741	7.12785	8.12828	2.1173	4.18841	5.18871	6.18847	7.18866	1.17801
4.73946	5.73943	6.73963	7.73904	1.72864	4.8165	5.81617	6.81637	7.81647	1.80584
5.12957	6.12939	7.12963	8.12981	2.1191	5.64967	6.64987	7.64977	8.64935	2.63893
4.15283	5.15291	6.15354	7.15323	1.14259	14.933	6.22947	7.22942	8.22896	2.21873
4.38748	5.388	6.38829	7.38818	1.37734	17.349	6.96361	7.9629	8.96349	2.95273
4.32358	5.32357	6.32334	7.32348	1.31264	17.7087	6.94906	7.9491	8.94854	2.93813
4.62577	5.62505	6.62519	7.62522	1.61485	21.4923	6.04899	7.04965	8.04906	2.0389
4.58471	5.58462	6.58528	7.58507	1.57446	14.8933	5.33575	6.33583	7.33632	1.32561

Table 7 Reported values of sensors 201 – 210

201	202	203	204	205	206	207	208	209	210
4.24268	5.24219	6.24221	7.24237	1.23188	4.5158	5.51435	6.51553	7.51487	2.67058
4.75668	5.7574	6.75749	7.75754	1.74657	4.87152	5.87442	6.87207	7.87238	1.595
5.69983	6.70018	17.3491	21.8283	2.68921	5.51331	6.51148	7.51112	8.51319	2.13546
5.80905	6.80939	15.5766	16.5871	2.79848	5.27817	6.2761	7.27482	8.2796	2.51145
4.99281	5.99295	18.101	14.3107	1.98207	5.59958	6.60165	7.60159	8.60339	2.68551
5.04782	6.04827	13.568	15.8527	2.03766	5.39406	6.39569	7.39398	8.39811	1.1378

Table 8 Reported values of sensors 1 – 10

1	2	3	4	5	6	7	8	9	10
5.59887	6.59941	7.59961	8.59896	2.58874	5.65474	6.65406	7.6545	8.65481	2.64405
4.81838	5.8182	6.81778	7.81776	1.8075	4.05023	5.05016	6.04997	7.0504	1.03954
4.44253	5.44257	6.44269	7.4425	1.43195	4.08135	5.08083	6.08096	7.08078	1.07069
4.28158	5.28115	21.1863	16.8539	11.4394	4.17511	5.17531	6.17549	7.17536	1.16472
5.88423	6.88499	18.8476	19.0108	11.8215	4.50881	5.50878	6.50887	7.50901	1.49846
5.06817	6.06755	13.6775	15.6218	11.6244	4.65855	5.65854	6.6592	7.65875	1.64833
5.21468	6.21418	18.6441	21.1322	11.5983	5.78057	6.78041	7.78031	8.78061	2.77021

6.4.2 Magnitude Test Graphs. X-axis represents time varying from 1 to 35sec and Y-axis represents Magnitude deviation. Magnitude Deviations of Some groups are shown in the following sections.

6.4.2.1 Magnitude Deviations of Sensors 126-130 Group. Deviations of sensors 126-130 with time is shown in Fig 15. Node 126 is showing highest deviation. Deviation Values of sensors 126-130 are shown in Table 9. Values that failed Magnitude test are shown in BOLDFACE.

6.4.2.2 Magnitude Deviations of Sensors 201-205 Group. Deviations of sensors 201-205 with time is shown in Fig 16. Nodes 203 and 204 are showing highest deviation. Deviation values of sensors 201-205 are shown in Table 10. Values that failed Magnitude test are shown in BOLDFACE.

6.4.2.3 Magnitude Deviations of Sensors 1-5 Group. Deviations of sensors 1-5 with time is shown in Fig 17. Nodes 3,4 and 5 are compromised nodes. Deviations values of sensors 1-5 are shown in Table 11. Values that failed Magnitude test are shown in BOLDFACE.

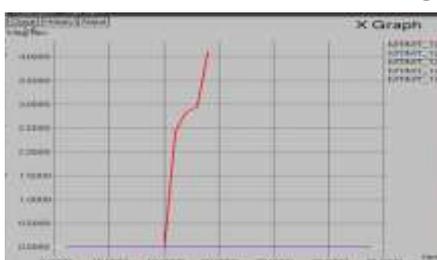


Figure 15. Magnitude deviations of Sensors 126-130

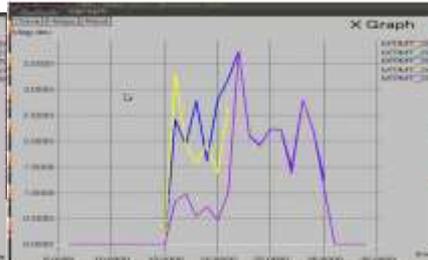


Figure 16. Magnitude deviations of sensors 201-205

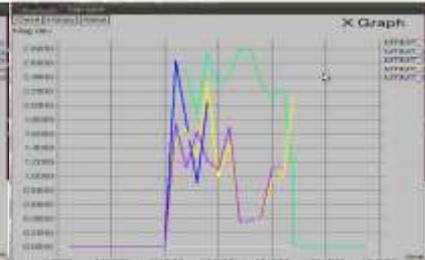


Figure 17. Magnitude deviations sensors 1-5

Table 9 Magnitude deviation values of sensors 126-130

Sensor 126	Sensor 127	Sensor 128	Sensor 129	Sensor 130
1.35E-05	3.33E-05	1.92E-05	3.33E-06	2.42E-05
6.45E-05	0.000209736	0.000121553	7.62E-05	7.65E-05
2.846487793	0.000221203	0.000408687	5.15E-05	7.99E-05
2.940038252	0.000218851	0.000159423	8.52E-05	9.98E-05
4.110752545	1.60E-05	5.79E-05	9.86E-05	2.48E-05

Table 10 Magnitude deviation values of sensors 201-205

Sensor 201	Sensor 202	Sensor 203	Sensor 204	Sensor 205
0.00012402	4.12E-05	0.000106185	2.11E-05	4.45E-05
0.000160967	5.64E-05	9.98E-05	3.63E-05	3.17E-05
0.836085755	0.835876649	2.412444804	3.281499965	0.836110312
0.972674327	0.972484309	1.938140754	1.940520447	0.972706404
0.540270728	0.540203736	2.763740968	1.579346848	0.540326553

Table 11 Magnitude deviation values of sensors 1-5

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
7.09E-05	5.89E-05	0.000127829	1.59E-05	1.40E-05
2.516240244	2.51629568	2.625554271	1.53900215	1.74069766
2.478610607	2.478357541	1.783954775	1.665862263	1.119392517
1.87040357	1.870510125	0.900416239	1.272814013	1.634958539

6.4.3 Shape Test Graphs. X-axis represents time varying from 1 to 35sec and Y-axis represents Shape deviation. Shape Deviations of Some groups are shown in the following sections. After removing compromised nodes one by one using characterization, deviation of the genuine sensors is gradually reduced. ‘-----’ in the Tables 12, 13 and 14 denotes that there is no considerable change in deviation values.

6.4.3.1 Shape Deviations of Sensors 126-130 Group. Deviations of sensors 126-130 with time is shown in Fig 18. Node 126 is the compromised node, which is detected and removed. Deviation Values of sensors 126-130 are shown in Table 12. Values that failed Shape test are shown in BOLDFACE.

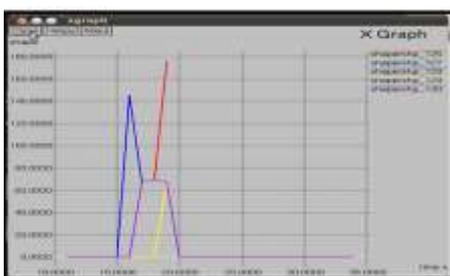


Figure 18. Shape deviations of Sensors 126-130

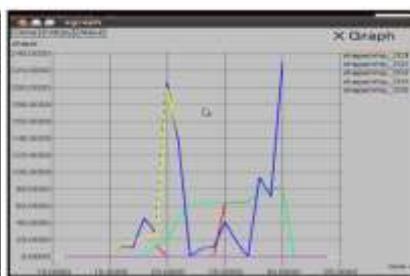


Figure 19. Shape deviations of sensors 201-205

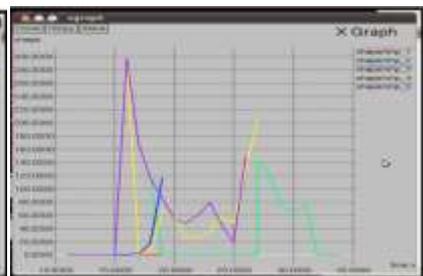


Figure 20. Shape deviations of sensors 1-5

6.4.3.2 Shape Deviations of Sensors 201-205 Group. Deviations of sensors 201-205 with time is shown in Fig 19. Nodes 203 and 204 are compromised nodes, which are detected and removed. Deviation Values of sensors 201-205 are shown in Table 13. Values that failed Shape test are shown in BOLDFACE.

6.4.3.3 Shape Deviations of Sensors 1-5 Group. Deviations of sensors 1-5 with time is shown in Fig 20. Nodes 3,4 and 5 are compromised nodes, which are detected and removed. Deviation Values of sensors 1-5 are shown in Table 14. Values that failed Shape test are shown in BOLDFACE.

Table 12 Shape deviation values of sensors 126-130

126	127	128	129	130
1.65E-05	1.54E-05	-----	4.38E-06	-----
1.64E-05	1.17E-05	7.53E-06	3.70E-06	5.16E-06
6.50E-06	8.28E-06	1.35E-05	3.35E-06	6.51E-06
7.18E-06	8.40E-06	0	6.17E-06	67.36099698
3.70E-06	-----	145.0064691	1.04E-05	69.536956
145.0064594	67.35908511	67.38849629	2.02E-05	67.36097652
67.38848337	-----	-----	6.78E-06	1.90E-06
69.55463568	8.65E-06	7.81E-06	7.80E-06	6.50E-07
176.106699	1.14E-05	0	67.36376917	4.55E-06
0	1.84E-06	3.03E-05	-----	-----

Table 13 Shape deviation values of sensors 201-205

201	202	203	204	205
7.23E-05	-----	28.46272821	0.000112086	-----
-----	0.000140299	205.1597015	5.82E-05	3.48E-06
2.08E-05	0.000208336	134.3838577	0.000135929	2.97E-06
0.000173254	-----	0	1.38E-05	0
3.66E-05	63.81393438	40.99932251	0.00022487	0
0	75.02005439	17.59129903	13.80509629	1.70E-06
63.6483309	81.26327979	0	18.76282955	1.60E-06
81.23963905	0	93.55296147	21.65300128	0
1.58E-06	1.37E-06	71.19815425	24.84098676	2.54E-06
-----	-----	229.6740635	199.4565964	1.86E-05

Table 14 Shape deviation values of sensors 1-5

1	2	3	4	5
-----	-----	3.46E-05	4.528435846	166.6827452
1.03E-05	297.396576	3.56E-05	66.55930759	117.0281543
5.90E-06	166.6827445	7.06E-06	58.91001041	85.62986275
295.9166878	117.0281539	3.11E-05	25.26485236	54.54841833
2.22E-05	4.76E-05	5.73E-06	25.54764332	48.26084512
9.18E-05	7.66E-05	0.172046639	36.45830003	60.39001082
0.000109238	0.000109238	2.003560799	58.93162759	79.11475223
1.41E-05	1.41E-05	17.5557588	48.79997176	43.60739501
0	0	116.7392391	129.9447745	19.41034435
-----	-----		204.7870369	148.8704306

6.4.4 Comparison Graphs. These are the graphs comparing 1. Existing System (In presence of injections) 2. Detection of Malicious data injections (Centralized approach) 3. Extended work (Distributed approach).

6.4.4.1 Remaining Node Energy. The remaining node energy of all sensors at the end of simulation in all existing, centralized and distributed architecture has been plotted in Fig 21 in which, the x-axis represents no.of nodes and y-axis represents energy values. As the detection is done at the

cluster head level itself in the Distributed architecture, it has more remaining node energy than centralized as well as existing systems.

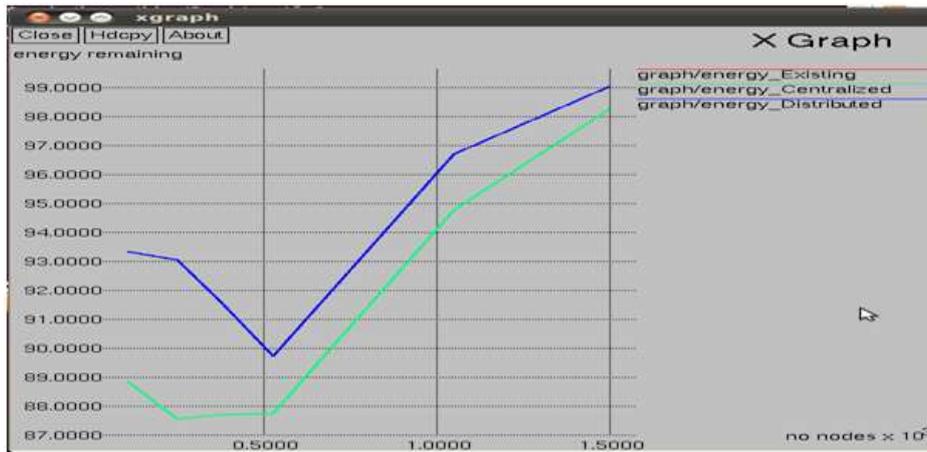


Figure 21. Remaining energy node graph of 1500 nodes

Energy values in joules are shown in Table 15.

Table 15. Remaining node energy values in Joules

Nodes	Existing System	Centralized System	Distributed System
105	88.847275214478088	88.847275214478088	93.336856113821753
250	87.54421268966162	87.54421268966162	93.047414870190948
375	87.707016934792065	87.707016934792065	91.6123671731184
525	87.761369372184362	87.761369372184362	89.718091244681588
1050	94.773528802887981	94.773528802887981	96.693384860179378
1500	98.255326077852303	98.255326077852303	99.054868934995142

6.4.4.2 Life Time of Network. The Life time graph of all the sensors in all existing, centralized and distributed architecture has been plotted in the above Fig 22, in which, the x-axis represents no.of nodes and y-axis represents life time in seconds. Distributed architecture being an in-network process, has better life time than centralized as well as existing systems.

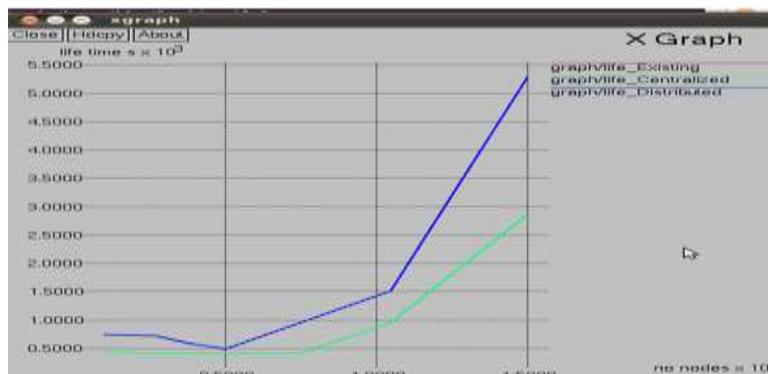


Figure 22. Life time graph of all the sensors (1500 nodes)

Life time values in seconds are shown in Table 16.

Table 16. Life time values in seconds

Nodes	Existing System	Centralized System	Distributed System
105	448.320934673	448.320934673	750.396522334
250	401.419828022	401.419828022	719.156962
375	406.73610087	406.73610087	596.115746027
525	408.542438452	408.542438452	486.291030098

1050	956.668431037	956.668431037	1512.12033713
1500	2865.86504018	2865.86504018	5290.2715667

6.4.4.3 Accuracy. The Accuracy graph of Estimation based frame work in all existing, centralized and distributed architecture has been plotted in Fig 23. Both the centralized and distributed architectures show 100% accuracy.

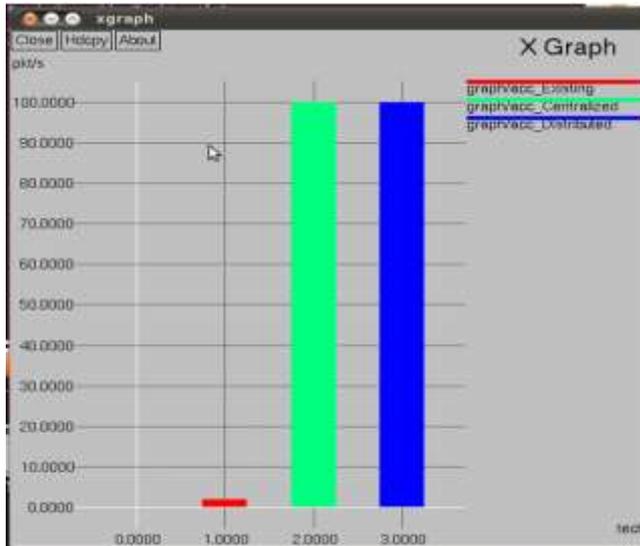


Figure 23. Accuracy graph of Estimation based Framework

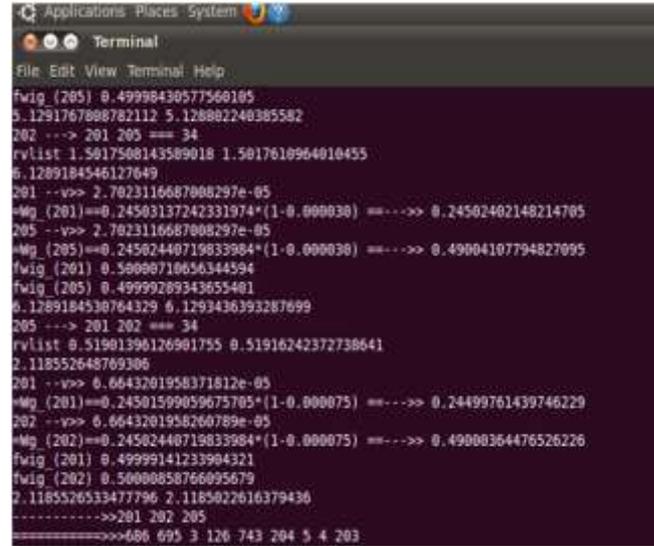


Figure 24. List of compromised nodes (values printed in the last line)

6.4.5 Terminal Output. Figure 24 shows compromised nodes, which are printed at the bottom of terminal.

VII. CONCLUSION AND FUTURE WORK

A typical wireless sensor network consists of several tiny and low-power sensors which use radio frequencies to perform distributed sensing tasks. Considered a scenario where an attacker gains full control of one or more sensors and can run arbitrary malware on them to fabricate new measurements and report them in place of the observed ones. The malicious data can be successfully detected and deleted at the base station.

Malicious injections detection system is implemented in centralized architecture, so in the proposed model, the base station only eliminates the malicious data. so the malicious data travels through the network node and it makes energy wastage in network. This work is enhanced to distributed architecture. As it is an in-network process, the process of detection of malicious injections is evenly distributed in the network. In order to avoid transmission of malicious data through the network nodes, to curtail the energy wastage and to enhance the lifetime of the network, the detection is done at the cluster head level itself by maintaining the accuracy using the LEACH characteristic. In this malicious data injection system, it is considered that the attacks may compromise the measurements even before they are transmitted, so in future work, malicious data injection system can be extended to detect the routing attacks which may occur during data transmission process.

VIII. ACKNOWLEDGMENT

Authors would like to thank all anonymous reviewers for their valuable comments. We also thank all technical staff of CSED, MVSREC for extending their support and giving valuable suggestions. And also thanks to parents for their support. Finally thanks to almighty god for being with us in each and every moment.

REFERENCES

- [1] T. He *et al.*, “Energy-efficient surveillance system using wireless sensor networks,” in *Proc. MobiSys*, 2004, pp. 270–283.
- [2] Otto, A. Milenković, C. Sanders, and E. Joránov, “System architecture of a wireless body area sensor network for ubiquitous health monitoring,” *J. Mobile Multimedia*, vol. 1, no. 4, pp. 307–326, Jan. 2005.
- [3] G. Werner-Allen *et al.*, “Deploying a wireless sensor network on an active volcano,” *Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar./Apr. 2006.
- [4] Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla, “SCUBA: Secure code update by attestation in sensor networks,” in *Proc. Workshop Wireless Security*, 2006, pp. 85–94.
- [5] D. Zhang and D. Liu, “DataGuard: Dynamic data attestation in wireless sensor networks,” in *Proc. IEEE/IFIP Int. Conf. DSN*, 2010, pp. 261–270.
- [6] T. Park and K. G. Shin, “Soft tamper-proofing via program integrity verification in wireless sensor networks,” *Trans. Mobile Comput.*, vol. 4, no. 3, pp. 297–309, May/June. 2005.
- [7] Sun, X. Shan, K. Wu, and Y. Xiao, “Anomaly detection based secure in-network aggregation for wireless sensor networks,” *Syst. J.*, vol. 7, no. 1, pp. 13–25, Mar. 2013.
- [8] Q. Zhang, T. Yu, and P. Ning, “A framework for identifying compromised nodes in wireless sensor networks,” *Trans. Inf. Syst. Secur.*, vol. 11, no. 3, pp. 1–37, Mar. 2008.
- [9] F. Liu, X. Cheng, and D. Chen, “Insider attacker detection in wireless sensor networks,” in *Proc. 26th IEEE INFOCOM, 2007*, pp. 1973–1945.
- [10] V. Hinds, “Efficient detection of compromised nodes in a wireless sensor network,” in *Proc. SpringSim, 2009*, Art ID. 95.
- [11] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, “On datacentric trust establishment in ephemeral ad hoc networks,” in *Proc. 27th IEEE INFOCOM*, 2008, pp. 1–11.
- [12] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, “Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection,” *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 2, pp. 169–183, Jun. 2012.
- [13] S. Ganeriwal, L. Balzano, M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” *Trans. Sensor Netw.*, vol. 4, no. 3, pp. 1–37, 2008.
- [14] W. Zhang, S. K. Das, and Y. Yonghe, “A trust based framework for secure data aggregation in wireless sensor networks,” in *Proc. 3rd Annu. IEEE SECON, 2006*, pp. 60–69.
- [15] Meena Malik, Dr. Yudhvir Singh, Anshu Arora “ Analysis of LEACH Protocol in Wireless Sensor Networks” *International Journal of Advanced Research in Computer Science and Software Engineering*. Volume 3, Issue 2, February 2013
- [16] Xiangning, F., and Yulin, S.. Improvement on LEACH protocol of wireless sensor network. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on* (pp. 260 - 264). IEEE. 2007
- [17] S. Tanachaiwiwat and A. Helmy, “Correlation analysis for alleviating effects of inserted data in wireless sensor networks,” in *Proc. MobiQuitous*, 2005, pp. 97–108.
- [18] Liu, X. Cheng, and D. Chen, “Insider attacker detection in wireless sensor networks,” in *Proc. 26th IEEE INFOCOM, 2007*, pp. 1973–1945.
- [19] Vittorio P. Illiano and Emil C. Lupu “Detecting Malicious Data Injections in Event Detection Wireless Sensor Networks” *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*, VOL. 12, NO. 3, SEPTEMBER 2015
- [20] Humaira Nishat, Vamsi Krishna K, Dr. D.Srinivasa Rao and Shakeel Ahmed, “Performance Evaluation of On Demand Routing Protocols AODV and Modified AODV (R-AODV) in MANETS” *International Journal of Distributed and Parallel Systems (IJDPS)* Vol.2, No.1, January 2011