Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-5, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Secure Scheme for Prevention of Malicious Node and Packet Drop in Wireless Sensor Network by Using PANEL System

Nilofar Nizammoddin Attar & Prajakta .A. Satarkar
CSE Department,SVERI COE ,Pandharpur

Abstract: *Large-scale sensor networks are deployed in numerous application domains, and the data they collect are used in decision-making for critical infrastructures. Data are streamed from multiple sources through intermediate processing nodes that aggregate information. A malicious adversary may introduce additional nodes in the network or compromise existing ones. Therefore, assuring high data trustworthiness is crucial for correct decision-making. Data provenance represents a key factor in evaluating the trustworthiness of sensor data. Provenance management for sensor networks introduces several challenging requirements, such as low energy and bandwidth consumption, efficient storage and secure transmission. Lightweight scheme to securely transmit provenance for sensor data is proposed.*

The technique relies on in-packet Bloom filters to encode provenance. It introduces efficient mechanisms for provenance verification and reconstruction at the base station. In addition, it extend the secure provenance scheme with functionality to detect packet drop attacks staged by malicious data forwarding node. The proposed technique both analytically and empirically, and the results prove the effectiveness and efficiency of the secure scheme in detecting packet forgery and loss attack.

Keywords: *sensor nodes ,forgery.*

1. Introduction:

Sensor networks are used in numerous application domains, such as cyber physical infrastructure systems, environmental monitoring, power grids, etc. Data are produced at a large number of sensor node sources and processed in network at intermediate hops on their way to a base station (BS) that performs decision-making. The diversity of data sources creates the need to assure the trustworthiness of data, such that only trustworthy information is considered in the decision process. Data provenance is an effective method to assess data trustworthiness, since it summarizes the history of ownership and the actions performed on the data. Recent research [1] highlighted the key contribution of provenance in systems where the use of untrustworthy data may lead to catastrophic failures (e.g., SCADA systems). Although provenance modeling, collection, and querying have been studied extensively for work flows and curated databases [2], [3], provenance in sensor networks has not been properly addressed.It investigate the problem of secure and efficient provenance transmission and processing for sensor networks, and we use provenance to detect packet loss attacks staged by malicious sensor nodes**.**

In a multi-hop sensor network, data provenance allows the BS to trace the source and forwarding path of an individual data packet. Provenance must be recorded for each packet, but important challenges arise due to the tight storage, energy and bandwidth constraints of sensor nodes. Therefore, it is necessary to devise a light-weight provenance solution with low overhead. Furthermore, sensors often operate in an untrusted environment, where they may be subject to attacks. Hence, it is necessary to address security requirements such as confidentiality, integrity and fresh-ness of provenance. The goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs. It propose a provenance encoding strategy whereby each node on the path of a data packet securely embeds provenance information within a Bloom filter (BF) that is transmitted along with the data. Upon receiving the packet, the BS extracts and verifies the provenance information. It also devise an extension of the provenance encoding scheme that allows the BS to detect if a packet drop attack was staged by a malicious node.

2. Objective:

1**.** Reduce energy consumption compared to other system.
2. Make less memory usage.
3. Analysis, detection and recovery from the malicious node in wireless network using PANEL system

.**3. Software Requirement**:

1.	Operating system	:	Ubuntu 10.04
2.	Tool	:	NS2.34
3.	Front End	:	Tool Command Language
4.	Coding Language	:	C/C++

4. Literature Survey:

Some of the recent and most relevant works are summarized below:

[1] Recent research H. Lim, Y. Moon, and E. Bertino highlighted the key contribution of provenance in systems where the use of untrustworthy data may lead to catastrophic failures (e.g., SCADA systems). Large number of application areas, like location-based services, transaction logs, sensor networks is qualified by uninterrupted data stream from many. Chasing of data provenance in extremely active circumstance is a crucial requirement, because data provenance is a key component in appraising data trustiness which is important for lots of application. Provenance handling of continuous data needs to cover various issues, admitting the storage efficiency, processing throughput, bandwidth conception and secure transmission.

[2] I. Foster, J. Vockler, M. Wilde, and Y. Zhao, gives information about provenance modeling, collection, and querying and shows studies extensively for workflows and curated databases .The Swift parallel scripting language allows for the speci_cation, execution and analysis of large-scale computations in parallel and distributed environments. It incorporates a data model for recording and querying provenance information. In this article it describes these capabilities and evaluates interoperability with other systems through the use of the Open Provenance Model. It describe Swift's provenance data model and compare it to the Open Provenance Model. It also describe and evaluate activities performed within the Third Provenance Challenge, which consisted of implementing a speci_c scienti_c workow, capturing and recording provenance information of its execution, performing provenance queries, and exchanging provenance information with other system.

[3] K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer [3], gave the basic terminologies used in provenance record in sensor networks Sensor network data has both historical and realtime value. Making historical sensor data useful, in particular, requires storage, naming, and indexing. Sensor data presents new challenges in these areas. Such data is location-specific but also distributed; it is collected in a particular physical location and may be most useful there, but it has additional value when combined with other sensor data collections in a larger distributed system. Thus, arranging location-sensitive peer-to-peer storage is one challenge. Sensor data sets do not have obvious names, so naming them in a globally useful fashion is another challenge. The last challenge arises from the need to index these sensor data sets to make them searchable. The key to sensor data identity is provenance, the full history or lineage of the data.

[4] Y. Simmhan, B. Plale, and D. Gannon showed that the emploment of separate transmission channels for data and provenance methology. Current provenance collection systems typically gather metadata on remote hosts and submit it to a central server. In contrast, several data-intensive scientific applications require a decentralized architecture in which each host maintains an authoritative local repository of the provenance metadata gathered on that host. The latter approach allows the system to handle the large amounts of metadata generated when auditing occurs at fine granularity, and allows users to retain control over their provenance records. The decentralized architecture, however, increases the complexity of auditing, tracking, and querying distributed provenance. We describe a system for capturing data provenance in distributed applications, and the use of provenance sketches to optimize subsequent data provenance queries. Experiments with data gathered from distributed workflow applications demonstrate the feasibility of a decentralized provenance management system and improvements in the Efficiency of provenance queries.

[5] R. Hasan, R. Sion, and M. Winslett, Traditional provenance security solutions use intensively cryptography and digital signatures and they employ

append-based data structures to store provenance, leading to prohibitive costs. With no provenance records will make the data highly suspicious and hence generate an alarm at the BS

The proposed technique do not consider denial of service attacks such as the complete removal of provenance, since a data packet with no provenance records will make the data highly suspicious Nevertheless, this system traces the source of a stream long after the process has completed .Hasan et al propose a chain model of provenance and ensure integrity and confidentiality through encryption, checksum and incremental chained signature mechanism.

[6] S. Madden, J. Franklin, J. Hellerstein, and W. Hong assume a multiple-round process of data collection. Each sensor generates data periodically, and individual values are aggregated towards the BS using any existing hierarchical (i.e., tree-based) dissemination scheme. Present the Tiny AGgregation (TAG) service for aggregation in low-power, distributed, wireless environments. TAG allows users to express simple, declarative queries and have them distributed and executed efficiently in networks of low-power, wireless sensors. It discusses various generic properties of aggregates, and show how those properties affect the performance of our in network approach. It includes a performance study demonstrating the advantages of our approach over traditional centralized, out-of-network methods, and discusses a variety of optimizations for improving the performance and fault-tolerance of the basic solution.

[7] K. Dasgupta, K. Kalpakis, and P. Namjoshi showed how the sequence number is attached to the packet by the data source, and all nodes use the same sequence number for a given round. Energy is one of the most important items to determine the network lifetime due to low power energy nodes included in the network. Generally, data aggregation tree concept is used to find an energy efficient solution. However, even the best aggregation tree does not share the load of data packets to the transmitting nodes fairly while it is consuming the lowest possible energy of the network. Therefore, after some rounds, this problem causes to consume the whole energy of some heavily loaded nodes and hence results in with the death of the network. In this paper, by using the Genetic Algorithm (GA), we investigate the energy efficient data collecting spanning trees to find a suitable route which balances the data load throughout the network and thus balances the residual energy in the network in addition to consuming totally low power of the network. Using an algorithm which is able to balance the residual energy among the nodes can help the network to withstand more and consequently extend its own lifetime.

[8] S. Sultana, E. Bertino, and M. Shehab showed the detail concept of distributed computing environment.the use of bloom filters in the distributed systems for provenance detection and its use. Malicious packet dropping attack is a major security threat to the data traffic in the sensor network, since it reduces the legal network throughput and may hinder the propagation of sensitive data. Dealing with this attack is challenging since the unreliable wireless communication feature and resource constraints of the sensor network may cause communication failure and mislead to the incorrect decision about the presence of such attack.

[9] L. Fan, P. Cao, J. Almeida, and A.Z. Broder introduces a counting bloom filter (CBF) associates a small counter with every bit, which is incremented/decremented Upon item insertion/deletion. Study the set reconciliation problem, in which each member of a node pair has a set of objects and seeks to deliver its unique objects to the other member. How could each node compute the set difference, however, is challenging in the set reconciliation problem. To address such an issue, we propose a lightweight but efficient method that only requires the pair of nodes to represent objects using a counting Bloom filter (CBF) of size $(O(d))$ and exchange with each other, where (d) denotes the total size of the set differences. A receiving node then subtracts the received CBF from its local one via minus operation proposed in this paper. The resultant CBF can approximately represent the union of the set differences and thus the set difference to each node can be identified after querying the resultant CBF.

[10] A. Kirsch and M. Mitzenmacher answers the approximate set membership queries, the distance-sensitive Bloom filter has been proposed by them. Transactional Memory (TM) is an alternative to conventional multithreaded programming to ease the writing of concurrent programs. In the context of unbounded TM, concurrent threads may use hardware signatures to record all the memory addresses issued inside a transaction to detect conflicts. Signatures are usually implemented as per-thread fixed hardware Bloom filters that summarize a very large amount of read and write memory addresses at the cost of false conflicts (detection of nonexisting conflicts). In this paper, to reduce the

probability of false conflicts, a novel signature design that exploits spatial locality is proposed. The design is based on new hash function mappings, so that nearby located addresses share some bits inserted in the filters. This is favorable particularly for large transactions that usually exhibit some amount of spatial locality. Besides, its implementation does not require extra hardware.

5. Existing system:

1. Data Packet Representation*:* It assumes a multiple-round process of data collection. Each sensor generates data periodically, and individual values are aggregated towards the BS using any existing hierarchical (i.e., tree-based) dissemination scheme [6]. A data path of D hops is represented as <nl; n1; n2; . . . ; nD >, where nl is a leaf node representing the data source, and node ni is i hops away from nl. Each non-leaf node in the path aggregates the received data and provenance with its own locally-generated data and provenance.

2. Provenance Encoding: For a data packet, provenance encoding refers to generating the vertices in the provenance graph and inserting them into the iBF. Each vertex originates at a node in the data path and represents the provenance record of the host node. A vertex is uniquely identified by the vertex ID. The VID is generated per-packet based on the packet sequence number (seq) and the secret key Ki of the host node. It uses a block cipher function to produce this VID in a secure manner. Thus for a given data packet, the VID of a vertex representing the node ni.When a source node generates a packet, it also creates a BF (referred to as ibf0), initialized to 0. The source then generates a vertex according to, inserts the VID into ibf0 and transmits the BF as a part of the packet.

3. Provenance Decoding at the BS: Not only the intermediate nodes, but also the BS stores and updates the latest packet sequence number for each data flow. Upon receiving a packet, the BS retrieves the preceding packet sequence (pSeq) transmitted by the source node from the packet header, fetches the last packet sequence for the flow from its local storage (pSeqb), and utilizes these two sequences in the process of provenance verification and collection.

3.1 Provenance verification. The BS first executes the provenance verification process upon receiving a packet. The BS knows 1) the current data path for the packet (decoded from the provenance ofthe previous packet in the flow), and 2) the preceding packet sequence number forwarded by each node in the path. In this context, the BS assumes that each node in the path saw and forwarded the same packet in the last round, and that this packet's sequence number is the same one as recorded at the BS. Thus the verification is bound to fail when pSeq and pSeqb do not match, which also indicates a possible packet loss and suffices to execute provenance collection process directly skipping the verification not match, which also indicates a possible packet loss and suffices to execute provenance collection process directly skipping the verification.

3.2 Provenance collection. Collection attempts to retrieve the nodes from the encoded provenance, confirm a packet loss and identify the malicious node that dropped the packet. It also distinguishes between the packet drop attack and other attacks that might have altered the iBF. Note that, in case of a path change, the new nodes can be easily learnt through an iteration of ibf membership testing over all the nodes. During provenance encoding, every new node in the path uses a special purpose packet identifier as the previous packet sequence and generates its VID Therefore, to retrieve the new nodes in the path, the decoding scheme at the BS should perform an ibf membership testing over all the nodes, where the VID for each node will be generated using the pre-specified previous packet identifier, along with the nodeID and the packet sequence number, seq[j].

The provenance collection scheme makes a list of potential vertices in the provenance graph through the ibf membership testing over all the nodes. For each node ni in the network, the BS creates the corresponding vertex (i.e., vi with VID vidi) .The BS then performs the membership query of vidi within ibf. If the algorithm returns true, the vertex is very likely present in the provenance, i.e., the host node ni is in the data path. Such an inference might introduce errors because of false positives (a node not on the route is inferred to be on the route). However the false positive probability obtained is very low.

4. Detect packet dropand identify malicious node:The secure provenance encoding scheme is extented to detect packet drop attacks and to identify malicious node (s). It assumes the links on the path exhibit natural packet loss and several adversarial nodes may exist on the path. For simplicity, It consider only linear data flow paths Also,It do not address the issue of recovery once a malicious node is detected. Existing techniques that are orthogonal to the detection scheme can be used, which may initiate

multipath routing or build a dissemination tree around the compromised nodes.

It augments the provenance encoding to use a packet acknowledgement that requires the sensors to transmit more meta-data. For a data packet, the provenance record generated by a node will now consist of the node ID and an acknowledgement in the form of a sequence number of the lastly seen (processed/forwarded) packet belonging to that use a pre-specified special purpose identifier, such as 0, as the previous packet sequence pSeqi. This addresses the case of routing path changes where a new node in the path can use this special identifier for encoding provenance. Moreover, if a node does not receive packets from a data flow for a long time, it can erase the previous packet information for that flow to reduce space overhead. The node can get updated and maintain this flow-specific record when it receives packets from that flow more frequently

6. Proposed System:

6.1. Overview of PANEL:

Position-Based Aggregator Node Election Protocol(**PANEL**) is introduced for wireless sensor networks. As its name indicates PANEL uses the geographical position information of the nodes to determine which of them should be the aggregators. Like other aggregator node election protocols, PANEL also ensures load balancing in the sense that each node is elected aggregator nearly equally frequently. The salient feature of PANEL that makes it novel and different from other aggregator node election protocols is that besides synchronous applications, PANEL also supports asynchronous applications.

PANEL assumes that the sensor nodes are deployed in a bounded area, and this area is partitioned into geographical clusters. For simplicity, in this paper, it is assume that the deployment area is a rectangle, and the clusters are equal sized squares. We emphasize, however, that the ideas behind PANEL are general, and PANEL could also be used for areas and cluster forms with more complex shapes.

The clustering is determined before the deployment of the network, and each sensor node is pre-loaded with the geographical information of the cluster which it belongs to.In our simplified case, each sensor node is pre-loaded with the coordinates of the lower-left corner of its cluster, as well as with the size d of the cluster. In addition, as we mentioned before, each node i is aware of its own geographical position $\vec{P}_i$**.**

Wireless sensor networks consist of a multitude of tiny sensor nodes capable for wireless communications and a few powerful base stations. The sensor nodes usually perform some monitoring task (e.g., measure various environmental parameters). The base stations collect sensor readings and forward them for further processing to a service centre. Based on how the sensor readings reach the base stations, synchronous and asynchronous sensor networks can be distinguished. In the synchronous case, the sensor readings are sent to the base stations in real-time using multi-hop wireless communications, where the sensor nodes cooperatively forward data packets on behalf of other sensor nodes towards the base stations. In the asynchronous case, the sensor readings are fetched by the base stations after some delay (e.g., once every day or week). In this case, the base stations are often mobile, and they physically approach the sensors in order to fetch their data through a single wireless hop.

PANEL assumes that the sensor nodes are deployed in a bounded area, and this area is partitioned into geographical clusters. For simplicity, in this paper, it is assume that the deployment area is a rectangle, and the clusters are equal sized squares. We emphasize, however, that the ideas behind PANEL are general, and PANEL could also be used for areas and cluster forms with more complex shapes.The clustering is determined before the deployment of the network, and each sensor node is pre-loaded with the geographical information of the cluster which it belongs to.In our simplified case, each sensor node is pre-loaded with the coordinates of the lower-left corner of its cluster, as well as with the size d of the cluster. In addition, as we mentioned before, each node i is aware of its own geographical position $\vec{P}_i$**.**

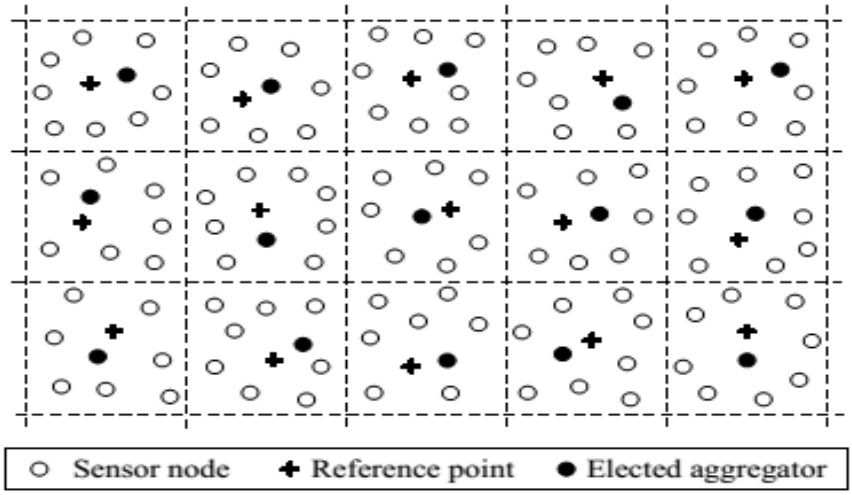


Fig: Geographical clustering in PANEL

At the beginning of each epoch, a reference point $\vec{R}_j$ is computed in each cluster *j* by every node in a completely distributed manner. In fact, the computation of the reference point depends only on the epoch number, and it can be executed by every node independently and locally. Once the reference point is computed, the nodes in the cluster elect the node that is the closest to the reference pointas the aggregator for the given epoch. The aggregator node election procedure needs communications within the cluster.

PANEL also includes a position-based routing protocol that is used in inter-cluster communications. As the nodes are aware of their geographical position, this seems to be a natural choice that does not result in additional overhead. The position-based routing protocol is used for routing messages from a distant base station or from a distant aggregator towards the reference point of a given cluster. Once the message enters the cluster, it is routed further towards the aggregator using the intra-cluster routing protocol based on the routing tables established during the aggregator node election procedure. Any position-based routing protocol can be integrated with PANEL. PANEL can also support reliable persistent data storage applications such as TinyPEDS. Reliability can be achieved by replicating the data aggregated by the aggregator nodes at other aggregator nodes. For this purpose, the aggregator nodes need to be able to communicate with each other.

The routing protocols of PANEL can support this by routing the messages containing the replicated data using PANEL's position-based inter-cluster routing protocol towards the reference point of the selected backup cluster, and then switching to the intra-cluster routing protocol of PANEL to deliver the data to the aggregator of that cluster. In PANEL, the reference points of the clusters are re-computed and the aggregator election procedure is re executed in each epoch. This ensures load balancing in the sense that each node of the cluster can become aggregator with nearly equal probability.

6.2. Methods:

6.2.1. Reference point computation

In PANEL, the aggregator election begins with the computation of a reference point $\vec{R}_j$ in each cluster *j*. The input of this computation is the current epoch number *e*, which is assumed to be known by every sensor. The reference point of cluster *j* is determined as,

$$\vec{R}_j = \vec{O}_j + Q,$$

Where $\vec{O}_j$ is the position of the lower-left corner of cluster *j*. The computation itself consists in calling a pseudo-random function *H* that maps *e* to a relative position *Q*inside the cluster. Formally, $H(e) = Q$, where $Q \in (-\delta d, d + \delta d) \times (-\delta d, d + \delta d)$, *d* is the size of the cluster. The pseudo-random function *H* can easily be implemented with a cryptographic hash function. Moreover, the pseudo randomness of *H* means that the outputs produced by *H* for the consecutive epoch numbers look as a sequence of random positions. This ensures the load balancing property of PANEL.Note that the above computation

can be executed by every sensor independently and locally.

6.2.2. Aggregator node election procedure:

Once the reference points are computed, the nodes start the aggregator node election procedure. Each node *i* sets a timer, the expiration time of which is proportional to the distance $D(\vec{P}_i, \vec{R}_j)$ between the node's position $\vec{P}_i$ and the reference point $\vec{R}_j$of its cluster. When this timer expires, the node broadcasts a message with maximum power in which it announces itself as the aggregator unless the node heard such an announcement from another node before its timer expired. The announcement message has the following format:

[*type* | *epoch* | *id* | *pos*]

where *type* is announcement, *epoch* is the current epoch number, and *id* and *pos* are the identifier and the position of the originator of the announcement, respectively.

6.2.3. Routing protocols:

In PANEL,there are two kinds of routing components:

1. intra-cluster routing protocol
2. Inter-cluster routing protocol.

The intra-cluster routing protocol is used to route a message to the aggregator of a given cluster if that message is already inside the cluster. This concerns, on the one hand, the messages that contain the measurements of the sensors in the cluster. On the other hand, the intra-cluster routing protocol is also used to route messages from a distant source to the current aggregator or to any of the past aggregators of the cluster once those messages have reached the cluster. These messages include queries originating from a distant base station and backup messages originating from aggregators of distant clusters. The intra-cluster routing protocol of PANEL can take advantage of the fact that the nodes within the cluster communicate during the aggregator election procedure.

The inter-cluster routing protocol is used to route messages to and from a distant cluster. These messages can be queries from and responses to a distant base station, as well as backup messages destined to distant aggregators that contain replicated data.

6.3. Pseudo code of PANEL:

Input:
identifier id_{self} and position $\vec{P}_{self}$ of the node executing the algorithm
parameters $\vec{O}_{self}$ and d of the cluster of the node executing the algorithm
current reference point $\vec{R}_{self}$ of the cluster and epoch number e_{now}
running time T of the algorithm
Output:
identifier id_{aggr} and position $\vec{P}_{aggr}$ of the elected aggregator node

set $id_{aggr} = id_{self}$;
set $\vec{P}_{aggr} = \vec{P}_{self}$;
set timer $t_0 = T$;
set timer $t_1 = f(D(\vec{P}_{self}, \vec{R}_{self}))$;
while timer t_0 is still active do
  wait until timer t_1 fires or an announcement m is received;
  case timer t_1 fired:
    broadcast [announcement | e_{now} | id_{self} | $\vec{P}_{self}$] with max power;
  case an announcement m = [announcement | e | id | $\vec{P}$] is received:
    if the pair (e, id) has been seen before then drop m;
    else if $e \neq e_{now}$ or $\vec{P} \notin square(\vec{O}_{self}, d)$ then drop m;
    else if $D(\vec{P}, \vec{R}_{self}) > D(\vec{P}_{aggr}, \vec{R}_{self})$ then drop m;
    else
      set $id_{aggr} = id$;
      set $\vec{P}_{aggr} = \vec{P}$;
      if timer t_1 is still active then cancel timer t_1;
      re-broadcast m with max power;
end while
output id_{aggr}, $\vec{P}_{aggr}$

When a node hears an announcement, it verifies if the originator of the announcement is closer to the reference point than the node known to be the closest so far (which can be the node itself if it has not heard any announcements yet). If so, then the node records the originator of the announcement as the candidate aggregator, and re-broadcasts the announcement. Moreover, if the node still has its timer active, then it cancels it. Otherwise, the node silently discards the announcement. Announcements that belong to other clusters are also discarded in order to limit the propagation of an announcement within the cluster that it is concerned with.

As the node that is the closest to the reference point sends its announcement first, there is a high chance that this will be the single announcement that is flooded inside the cluster. This means that in most cases, each node re-broadcasts a single message during the aggregator election procedure. In some cases, however, depending on the topology of the network, it may happen that more than one nodes send their announcements. In those cases, only the announcement originated by the node that is the closest to the reference point will "survive", meaning that only that announcement will be received and recorded by every node in the cluster.

After some predefined time *T*, the aggregator node election phase is closed, and each node considers the recorded candidate aggregator as the aggregator for the current epoch. The value of *T* depends on the time needed for a flooded message to cover the largest possible distance within the cluster. This ensures that at the end of the aggregator election phase, each node must have received the announcement of the future aggregator.

6.4. System Architecture:

In the architecture it describes the node placement and the simulation process done in this regard. In node placement side the mobile sink activity and the target length to be covered by the mobile sink is given. This process was done by sending a frequent beacon signal to identify the location of sensor nodes in the network. After identifying the node the path is selected to send the sensed data to the destination. Every nodes activities are monitored to select the cluster head for every group based on communication. Then the data are collected by the CH (cluster head). The data must be collected within the time given. The output is measured based on number of packets arrived and the energy used for communication. This was given by simulation process.

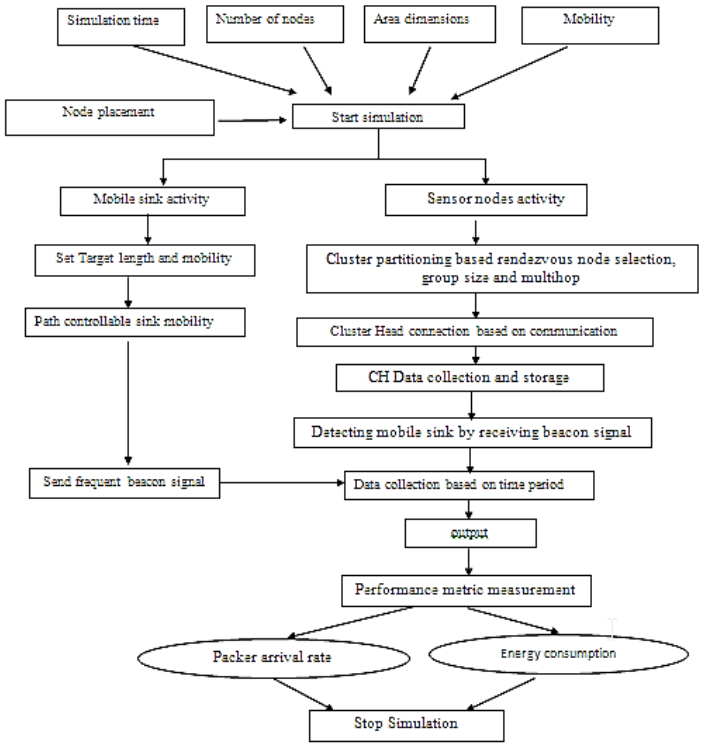
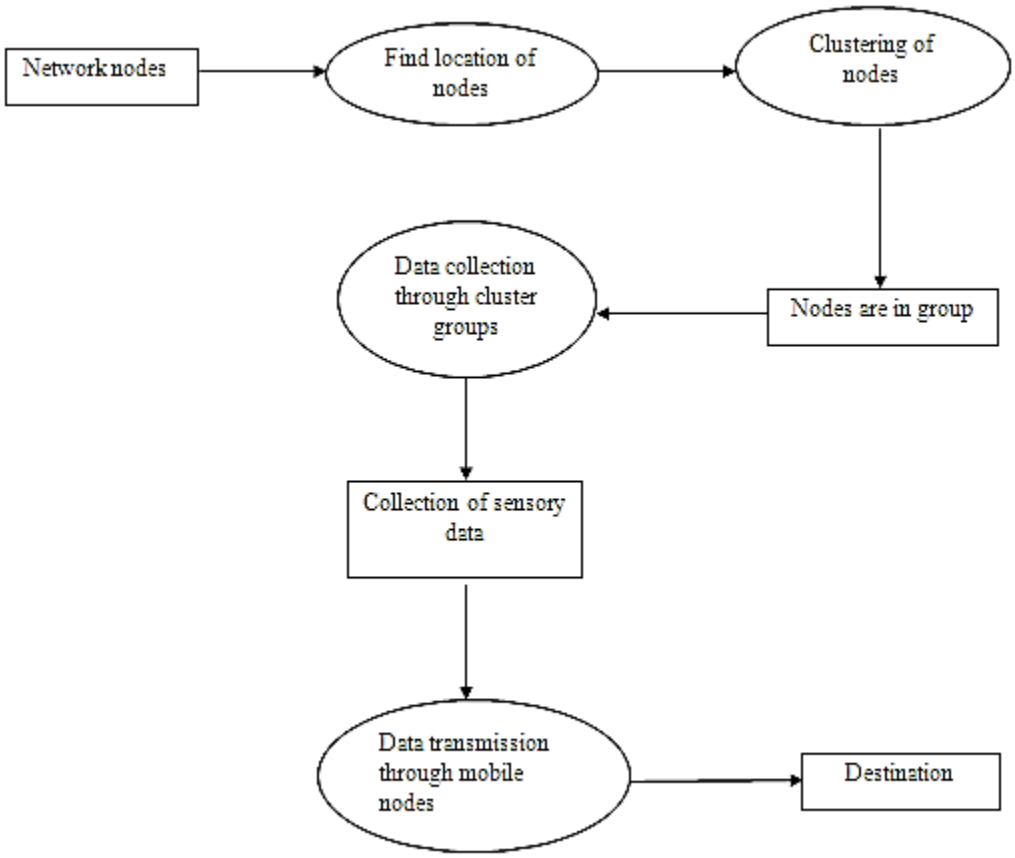


Fig: System Architecture

6.5. Data flow diagram:



Conclusion:

With our proposed system conclusion is made that the PANEL protocol is more efficient than other techniques for the detection of malicious node in terms of memory, time and energy usage. Therefore there is good future scope for this technique for making it much more efficient in solving the problems of wireless sensor network

References:

1. Andreas Merentitis, Nektarios Kranitis,Antonis Paschalis and Dimitris Gizopoulos,"Low Energy Online Self-Test of Embedded Processors in Dependable WSN Nodes" IEEE transactions on dependable and secure computing, vol. 9, no. 1, january/february 2012,pp.86-100.
2. Charalampos Konstantopoulos, Grammati Pantziou, Damianos Gavalas,Aristides Mpitziopoulos, and Basilis Mamalis,"A Sensor-Based Approach Enabling Energy-Efficient Sensory Data Collection with le Sinks" IEEE transactions on parallel and distributed systems, vol. 23, no. 5, May 2012, pp.809-817.
3. Degan Zhang,Guang Li,Ke Zhen, Xuechao Ming and Zhao-Hua Pan," An Energy-Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks" IEEE transactions on industrial informatics, vol. 10, no. 1, february 2014,pp.766-773.
4. Guoliang Xing,Tian Wang, Zhihui Xie, and Weijia Jia,"Sensor Planning in Wireless Sensor Networks with le Elements" IEEE transactions on le computing, vol. 7, no. 12, Dec 2008, pp.1430-1443.

5. Hana Besbes, George Smart,Dujdow Buranapanichkit,Christos Kloukinas, and Yiannis Andreopoulos," Analytic Conditions for Energy Neutrality in Uniformly-Formed Wireless Sensor Networks" IEEE transactions on wireless communications, vol. 12, no. 10, october 2013,pp.4916-4931.
6. Issa M. Khalil,"ELMO: Energy Aware Local Monitoring in Sensor Networks" IEEE transactions on dependable and secure computing, vol. 8, no. 4, july/august 2011,pp.523-536.
7. Jiajia Liu, Xiaohong Jiang, Hiroki Nishiyama and Nei Kato,"On the Delivery Probability of Two-Hop Relay MANETs with Erasure Coding" IEEE transactions on communications, vol. 61, no. 4, April 2013, pp.1314-1326.
8. Kashif Saleem, Norsheila Fisal and Jalal Al-Muhtadi," Empirical Studies of Bio-Inspired Self-Organized Secure Autonomous Routing Protocol" IEEE sensors jouSNal, vol. 14, no. 7, july 2014, pp.2232-2239.
9. Ljubica Blazevic, Jean-Yves Le Boudec and Silvia Giordano, "A Location-Based Routing Method for le Ad Hoc Networks" IEEE transactions on le computing, vol. 3, no. 4, October-December 2004, pp.1-15.
10. Marios Gatzianas and Leonidas Georgiadis, "A Distributed Algorithm for Maximum Lifetime Routing in Sensor Networks with le Sink" IEEE transactions on wireless communications, vol. 7, no. 3, March 2008, pp.984-994.
11. Oualid Demigha, Walid-Khaled Hidouci, and Toufik Ahmed," On Energy Efficiency in Collaborative Target Tracking in Wireless Sensor Network: A Review" IEEE communications surveys & tutorials, vol. 15, no. 3, third quarter 2013,pp.1210-1222.
12. Özgür B. Akan,and Ian F. Akyildiz," Event-to-NodeReliable Transport in Wireless Sensor Networks" IEEE/ACM transactions on networking, vol. 13, no. 5, october 2005,pp.1003-1016.