Design of A Secure Scheme employing In-Packet Bloom Filter for Detecting Provenance Forgery and Packet Drop Attacks in WSN

Rohit D. Hedau[1], Dr. Pankaj Agrawal[2]
[1]Student , Department of Electronics and Comm. Engg., G. H. R.A.E.T., Nagpur, India
[2]H.O.D., Department of Electronics and Comm. Engg., G. H. R.A.E.T., Nagpur, India

ABSTRACT

Lots of application are inherently wireless sensor network based wherein they sense the desired parameter and forward the data to the base station to be aggregated abd put forward to suitable use as and when desired. In such a case it is always a possibility that a malicious adversary may introduce additional nodes in the network or compromise existing ones. It is therefore highly crucial that high data trustworthiness is high so that correct decisions can be taken. Data provenance becomes critical in evaluating the trustworthiness of sensor data. Low energy and bandwidth consumption, efficient storage and secure transmission are several challenges required to be catered by provenance management .In this work we have been able to present a scheme to securely transmit provenance for sensor data. The scheme employs in packet bloom filter for providing provenance encoding . The provenance mechanism and reconstruction is taken care by the base station . The provenance scheme has been able to detect packet drop attacks staged by malicious data forwarding nodes.

I. INTRODUCTION

Sensor networks have made their presence in lots of domains such as cyber physical infrastructure systems, environmental monitoring, power grids, etc. large amount of data is generated and forwarded towards the base station. Trustworthiness of data is a big issue as the data is being generated and forwarded from diverse sensor networks .Data provenance is an effective method to assess data trustworthiness, since it summarizes the history of ownership and the actions per- formed on the data. Recent research highlighted the key contribution of provenance in systems where the use of untrustworthy data may lead to catastrophic failures. Although provenance modeling, collection, and querying have been studied extensively for workflows and curate databases, provenance in sensor networks has not been properly addressed. We will investigate the problem of secure and efficient provenance transmission and processing for sensor networks, and we will 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 freshness of provenance. Our goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs. We will 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. There is need to 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. As opposed to existing research that employs separate transmission channels for data and provenance, we only require a single channel for both. Furthermore, traditional provenance security solutions use intensively cryptography and digital signatures, and they employ append-based data structures to store provenance, leading to prohibitive costs. In contrast, we will use only fast message authentication code (MAC) schemes and Bloom filters, which are fixed-size data structures that compactly represent provenance. Bloom filters make efficient usage of bandwidth, and they yield low error rates in practice.

II. LITERATURE SURVEY

Salmin Sultana , Gabriel Ghinita , Elisa Bertino , Mohamed Shehab in their work "A Lightweight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless sensor Networks" have proposed a novel lightweight scheme to securely transmit provenance for sensor data. The proposed technique relies on in-packet Bloom filters to encode provenance and it introduces efficient mechanisms for provenance verification and reconstruction at the base station. In addition, the scheme is armed with functionality to detect packet drop attacks staged by malicious data forwarding nodes.

Sankardas Roy, Mauro Conti, Sanjeev Setia, and Sushil Jajodia. In their work "Secure Data Aggregation in Wireless Sensor Networks" have presented a novel lightweight verification algorithm by which the base station can determine if the computed aggregate (predicate Count or Sum) includes any false contribution. Thorough theoretical analysis and extensive simulation study show that our algorithm outperforms other existing approaches.

Fan Ye, Haiyun Luo, Songwu Lu, Lixia Zhang in their work "Statistical En-route Filtering of Injected False Data in Sensor Networks"have presented a Statistical En-route Filtering (SEF) mechanism that can detect and drop such false reports. SEF requires that each sensing report be validated by multiple keyed message authentication des (MACS), each generated by a node that detects the same event. As the report is forwarded, each node along the way verifies the correctness of the MACS and drops those with invalid macs at earliest points. The sink further filters out remaining false reports that escape the en-route filtering. SEF exploits the network scale to determine the truthfulness of each report through collective decision-making by multiple detecting nodes and collective false-report detection by multiple forwarding nodes. Their analysis and simulations show that, with an overhead of 14 bytes per report, SEF is able to dmp %90% injected false reports by a compromised node within 10 forwarding hops, and reduce energy consumption by 50% or more in many cases.

Salmin Sultana, Mohamed Shehab in their work "Secure Provenance Transmission for Streaming Data" have proposed a novel approach to securely transmit provenance for streaming data (focusing on sensor network) by embedding provenance into the inter packet timing domain while addressing the above mentioned issues. As provenance is hidden in another host-medium, our solution can be conceptualized as watermarking technique. However, unlike traditional watermarking approaches, we embed provenance over the inter packet delays (IPDs) rather than in the sensor data themselves, hence avoiding the problem of data degradation due to watermarking. Provenance is extracted by the data receiver utilizing an optimal threshold-based mechanism which minimizes the probability of provenance decoding errors. The resiliency of the scheme against outside and inside attackers is established through an extensive security analysis. Experiments show that our technique can recover provenance up to a certain level against perturbations to inter-packet timing characteristics.

On the basis of literature survey carried out we observe that recent research highlighted the key contribution of provenance in systems where the use of untrustworthy data may lead to catastrophic failures. Although provenance modeling, collection, and querying have been studied extensively for workflows and curated databases, provenance in sensor networks has not been properly addressed.

Traditional provenance security solutions use intensively cryptography and digital signatures, and they employ append-based data structures to store provenance, leading to prohibitive costs.

Existing research employs separate transmission channels for data and provenance.

There is a need to design a provenance encoding and decoding mechanism that satisfies such security and performance needs. Similarly there is a need of 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.

III. METHODOLOGY

3.1 Light weight secure provenance scheme for wireless sensor network

This section tries to present the methodology for implementing provenance management for streaming data focusing on WSNs. We try to examine a mechanism which generates and transmits provenance in a distributed setting where a source node generates the data and the intermediate node(s) towards the BS may process the in-transit data. A possible approach to the problem could be based on traditional security solutions like encryption, digital signature, and message authentication code (MAC). In a digital signature (or MAC) based mechanism, each party which is a stake holder in the data processing would affix its information to data and sign it to guarantee authenticity. Apart from this an approach base on encryption and incremental chained signature can be used for secure document provenance . This approach however brings in the difficulty of the provenance becoming larger than the data itself leading to inefficient usage of the transmission channel. Encryption/signature/MAC based mechanisms still are not capable to tackle this problem .Thus to address the above challenge we present two techniques - (i) a watermarking scheme for per-flow provenance encoding and decoding over the inter-packet delays (IPD), (ii) a per-packet provenance scheme using IBF. Different WSN applications may prefer one solution over the other depending on the network data rates.

3.2 Background and System Model

The system basically consists of the network, data and provenance models used. This section also deals with the details of the threat model and security requirements. Finally, it discusses the fundamental properties and operations of Bloom filters.

3.2.1 Network Model. We consider a multihop wireless sensor network, consisting of a number of sensor nodes and a base station (BS) that collects data from the network. The network is modeled as a graph G(N, L), where $N = \{n_i|, 1\ i\ |N|\}$ is the set of nodes, and L is the set of links, containing an element $l_{i,j}$ for each pair of nodes n_i and n_j that are communicating directly with each other.

3.2.2 Data Model. We assume a multiple-round process of data collection. Each sensor node generates data periodically, and individual values are routed and aggregated towards the BS using any existing hierarchical (i.e., tree-based) dissemination scheme, e.g., [5]. A data path of p hops is represented as $< n_l, n_1, n_2, ..., n_p >$, where n_l is a leaf node representing the data source, and node n_i is i hops away from n_l. Each non-leaf node in the path aggregates the received data and provenance with its own locally-generated data and provenance.

Each data packet contains a unique packet sequence number, a data value, time stamp, and provenance. The sequence number is attached to the packet by the data source, and all nodes use the same sequence number for a given round [77].

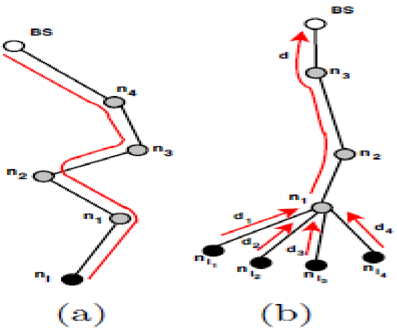


Figure 1. Provenance graph for a sensor network.

Depending on the solution approach considered, the timestamp/sequence number integrity is ensured through message authentication codes (MAC).

3.2.3Provenance Model. We consider node-level provenance, which encodes the nodes that are involved at each step of data processing. This representation has been used in previous research for trust management [6] and for detecting selective forwarding attacks [7].

Given a data packet d, its provenance is modeled as a directed acyclic graph G(V, E) where each vertex v 2 V is attributed to a specific node HOST (v) = n and represents the provenance record (i.e. nodeID) for that node. Each vertex in the provenance graph is uniquely identified by a vertex ID (VID) which is generated by the host node using cryptographic hash functions. The edge set E consists of directed edges that connect sensor nodes.

Threat Model and Security Objectives. We assume that the BS is trusted, but any other arbitrary node may be malicious. An adversary can eavesdrop and perform traffic analysis anywhere on the path and can get hold of critical information . Secondly the adversary may also be able to deploy a few malicious nodes, as well as compromise a few legitimate nodes by capturing them and physically overwriting their memory. We are concerned the following objectives:
Confidentiality: An adversary cannot gain any knowledge about data prove-nance by analyzing the IPDs or the contents of a packet. Only authorized parties (e.g., the BS) can process and check the integrity of provenance.

Integrity: An adversary, acting alone or colluding with others, cannot add or remove non-colluding nodes from the provenance of benign data (i.e. data generated by benign nodes) without being detected.
Freshness: An adversary cannot replay captured data and provenance without being detected by the BS.
However, an adversary may increase network jitter in a way that the recorded IPD at the BS is much larger than the desired value. Such an attack is intended to destroy the embedded provenance. We successfully recover provenance against the insertion attack but survive the deletion attack to a certain extent.

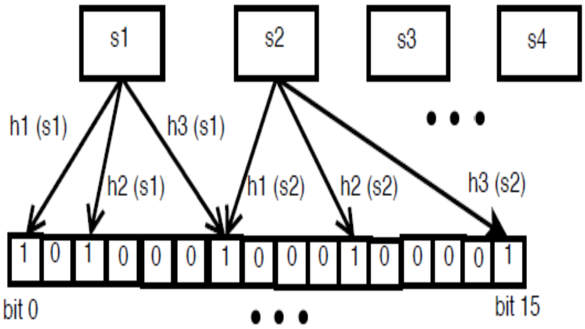


Figure 2. A Bloom filter with n = 4, m = 16 and k = 3.

Bloom Filters (BF). A Bloom filter is a space-efficient data structure for probabilistic representation of a set of items $S = \{s_1, s_2, ..., s_n\}$ using an array of m bits with k independent hash functions $h_1, h_2, ..., h_k$. The output of each hash function h_i maps an item s uniformly to the range [0, m-1] and is interpreted as an index point-ing to a bit in a m-bit array. Hence, the BF can be represented as $\{b_0, . . . , b_{m-1}\}$. Initially each of the m bits is set to 0.

To insert an element $s \in S$ into a BF, s is hashed with all the k hash functions producing the values $h_i(s)(1\leq i\leq k)$. The bits corresponding to these values are then set to 1 in the bit array. Figure 5.2 illustrates an example of BF insertion. To query the membership of an item s' within S, the bits at indices $h_i(s')$ $(1\leq i\leq k)$ are checked. If any of them is 0, then certainly $s' \in S$. Otherwise, if all of the bits are set to 1, $s' \in S$ with high probability. There exists a possibility of error which arises due to hashing collision that makes the elements in S collectively causing indices $h_i(s')$ being set to 1 even if $s' \notin S$. This is called a false positive. Note that, there is no false negative in the BF membership verification.

The cumulative nature of BF construction inherently supports the aggregation of BFs of a same kind, by performing bitwise-OR between the bitmaps.

3.2.4Provenance Encoding

After generating a data packet, the source node marks it with the generation time and ensures the integrity of the timestamp with a MAC. The MAC is computed using the node specific secret key K_i. The next L_p data packets generated by the node, more specifically, the sequence of L_p IPDs is the medium where we hide the provenance of the packets. We denote the set of IPDs by DS = { [1], [2], ..., [L_p] }, where j[] represents the IPD between j-th and (j+1)-th data packet. The data source encodes a bit of its PN sequence over each IPD. Throughout the transmission of a packet towards the BS, each intermediate node also encodes 1-bit of provenance information over the associated IPD. Hence, an IPD recorded at the BS carries the sum of 1-bit information from each node in the path. The process also uses the secret K_i and a locally generated random number α_i (known as impact factor). The BS only knows the distribution of the α_i's. The process a node n_i follows to encode a bit of PN sequence over an IPD is summarized below:

1) Generation of Delay Perturbations:
2) Selection of a Delay Perturbation:
3) Provenance Embedding:
   a) Simple Provenance Embedding
   b) Aggregate Provenance Embedding

3.2.5 Provenance Decoding

The provenance retrieval algorithm recovers provenance using the secret parame-ters including the keys $\{K_1, K_2, ..., K_n\}$, the PN length L_p, and the optimal threshold T ⇤. The threshold, corresponding to the network diameter and PN length, is calculated once after the deployment of the network.

IV. RESULT AND DISCUSSION

NETWORK FORMATION

The simulation work has been done with the Network Simulator ns-2, Version 2.34. Network formation is an aspect of creating nodes of network and transmit data. Network of 100 nodes is created using network simulator for wireless sensor network.

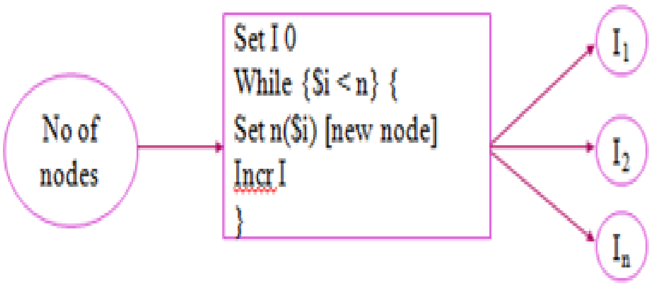
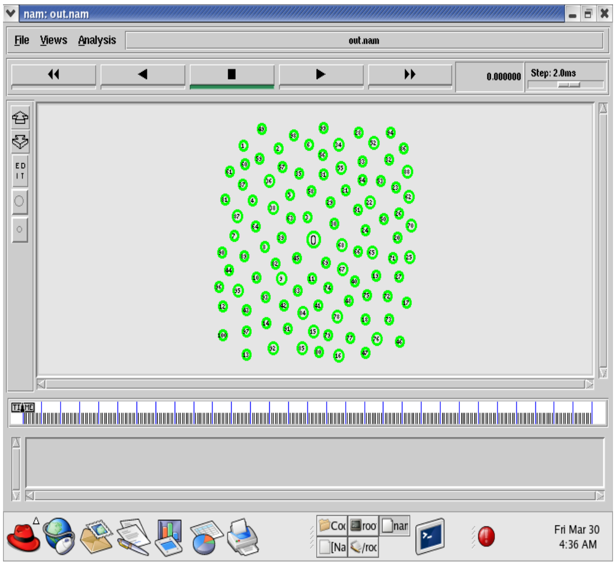


Figure 3. figure showing the method of network formation



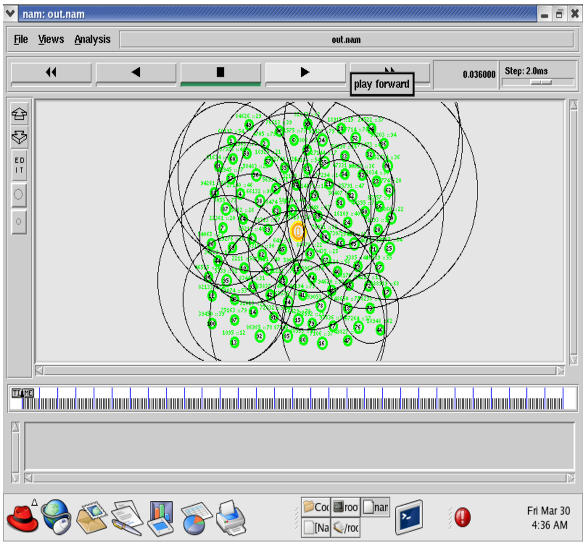
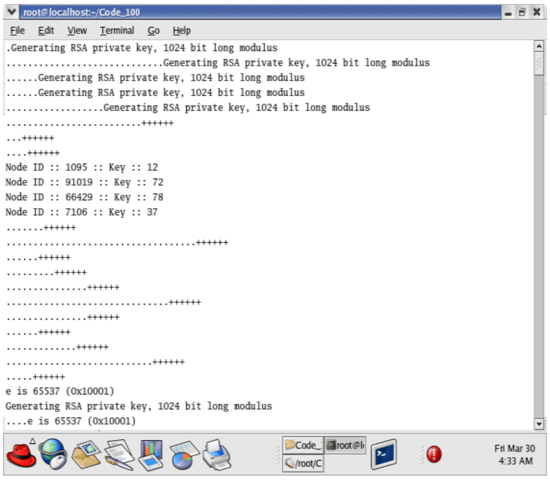


Figure 4. figure showing network formation

SECURITY IMPLEMENTATION

Encryption decryption is done by SHA Algorithm. Detection of misbehavior nodes using Security Packet, then send communication between source to destination node. SHA is an algorithm used by modern computers to encrypt and decrypt messages. If the authentication is successful then it sends data packet through the Reliable routing path. SHA provides end-to-end confidentiality and hop-by-hop authentication.

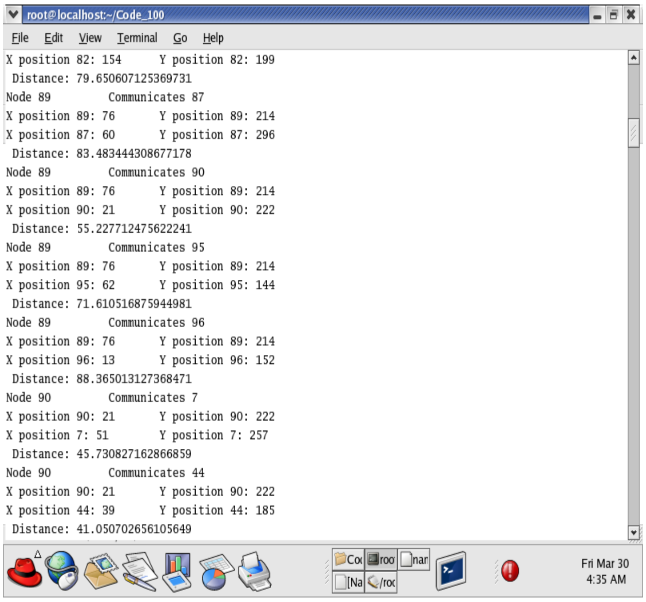


```
.Generating RSA private key, 1024 bit long modulus
.............................Generating RSA private key, 1024 bit long modulus
......Generating RSA private key, 1024 bit long modulus
......Generating RSA private key, 1024 bit long modulus
..................Generating RSA private key, 1024 bit long modulus
.........................++++++
...++++++
....++++++
Node ID :: 1095 :: Key :: 12
Node ID :: 91019 :: Key :: 72
Node ID :: 66429 :: Key :: 78
Node ID :: 7106 :: Key :: 37
.......++++++
...................................++++++
......++++++
.........++++++
...............++++++
..............................++++++
...............++++++
......++++++
.............++++++
...........................++++++
.....++++++
e is 65537 (0x10001)
Generating RSA private key, 1024 bit long modulus
....e is 65537 (0x10001)
```

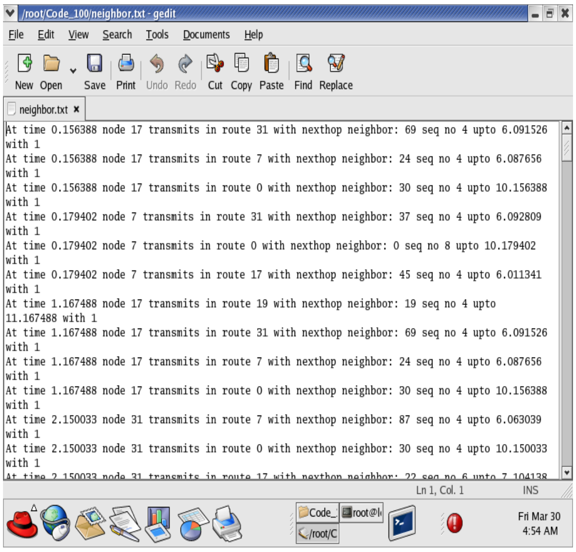
Figure 5. figure showing the security implementations

NEIGHBOR DISCOVERY

When a source node needs to find a route to a destination, it starts a route discovery process, based on flooding, to locate the destination node. Upon receiving a route request (RREQ) packet, intermediate nodes update their routing tables for a reverse route to the source. Similarly, the forward route to the destination is updated upon reception of a route reply (RREP) packet originated either by the destination itself or any other intermediate node that has a current route to the destination.



```
X position 82: 154      Y position 82: 199
 Distance: 79.650607125369731
Node 89          Communicates 87
X position 89: 76       Y position 89: 214
X position 87: 60       Y position 87: 296
 Distance: 83.483444308677178
Node 89          Communicates 90
X position 89: 76       Y position 89: 214
X position 90: 21       Y position 90: 222
 Distance: 55.227712475622241
Node 89          Communicates 95
X position 89: 76       Y position 89: 214
X position 95: 62       Y position 95: 144
 Distance: 71.610516875944981
Node 89          Communicates 96
X position 89: 76       Y position 89: 214
X position 96: 13       Y position 96: 152
 Distance: 88.365013127368471
Node 90          Communicates 7
X position 90: 21       Y position 90: 222
X position 7: 51        Y position 7: 257
 Distance: 45.730827162866859
Node 90          Communicates 44
X position 90: 21       Y position 90: 222
X position 44: 39       Y position 44: 185
 Distance: 41.050702656105649
```


/root/Code_100/neighbor.txt - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
neighbor.txt
Ln 1, Col. 1
INS
Code_
root@l
/root/C
Fri Mar 30
4:54 AM

```
At time 0.156388 node 17 transmits in route 31 with nexthop neighbor: 69 seq no 4 upto 6.091526
with 1
At time 0.156388 node 17 transmits in route 7 with nexthop neighbor: 24 seq no 4 upto 6.087656
with 1
At time 0.156388 node 17 transmits in route 0 with nexthop neighbor: 30 seq no 4 upto 10.156388
with 1
At time 0.179402 node 7 transmits in route 31 with nexthop neighbor: 37 seq no 4 upto 6.092809
with 1
At time 0.179402 node 7 transmits in route 0 with nexthop neighbor: 0 seq no 8 upto 10.179402
with 1
At time 0.179402 node 7 transmits in route 17 with nexthop neighbor: 45 seq no 4 upto 6.011341
with 1
At time 1.167488 node 17 transmits in route 19 with nexthop neighbor: 19 seq no 4 upto
11.167488 with 1
At time 1.167488 node 17 transmits in route 31 with nexthop neighbor: 69 seq no 4 upto 6.091526
with 1
At time 1.167488 node 17 transmits in route 7 with nexthop neighbor: 24 seq no 4 upto 6.087656
with 1
At time 1.167488 node 17 transmits in route 0 with nexthop neighbor: 30 seq no 4 upto 10.156388
with 1
At time 2.150033 node 31 transmits in route 7 with nexthop neighbor: 87 seq no 4 upto 6.063039
with 1
At time 2.150033 node 31 transmits in route 0 with nexthop neighbor: 30 seq no 4 upto 10.150033
with 1
At time 2.150033 node 31 transmits in route 17 with nexthop neighbor: 22 seq no 6 upto 7.104138
```

Figure 6. Screen shot showing neighbor discovery

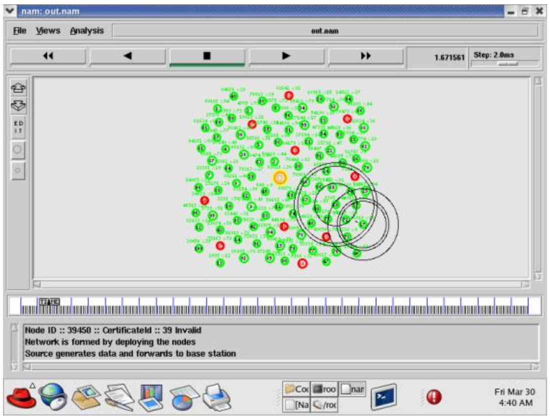
DATA FORWARDING

Route request is sent to all the intermediate nodes between source and destination. Route discovery for shortest and fresh path is done .When a source node needs to find a route to a destination, it starts a route discovery process, based on flooding, to locate the destination node.

Upon receiving a route request (RREQ) packet, intermediate nodes update their routing tables for a reverse route to the source. After reaches the destination node- Sends Route reply packets to source node. The data is transmitted from source node to destination node through energy efficient intermediate nodes and in case of failure the route discovery is again initiated.

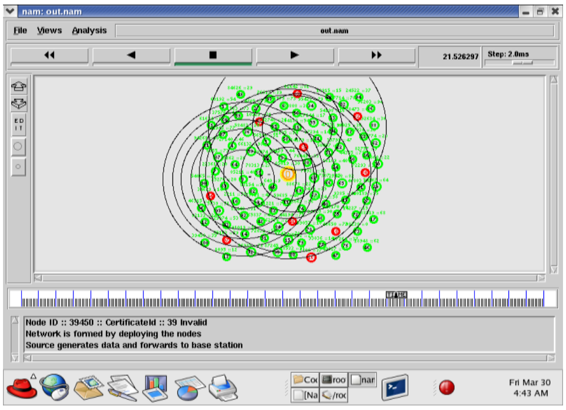
Anomalous Behavior Detection

Route request is sent to all intermediate nodes between source S and destination D. Route discovery for shortest and freshest path is carried out using Secure AODV Protocol. The neighbor list is checked so as to check anomalous behavior using security packets and then the communication is initiated between the source and the destination



Node ID :: 39450 :: CertificateId :: 39 Invalid
Network is formed by deploying the nodes
Source generates data and forwards to base station

Figure 7. showing screen shot for data



Node ID :: 39450 :: CertificateId :: 39 Invalid
Network is formed by deploying the nodes
Source generates data and forwards to base station

Figure 8. showing screen shot Anomalous forwarding Behavior detection

As seen in figure 5.5 we observe the network which is formed by deploying the nodes and the source generates the data and forwards to the base station .This figure show the behavior of the nodes while forwarding the data. In figure 5.6 we observe that nodes are transmitting the data and some node go on with anomalous behavior

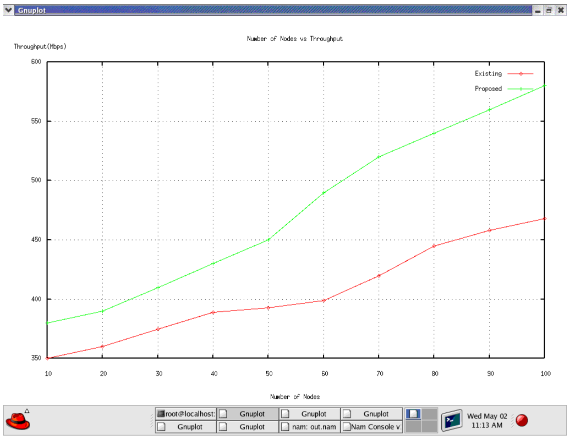


Figure 9. figure showing screenshot for throughput

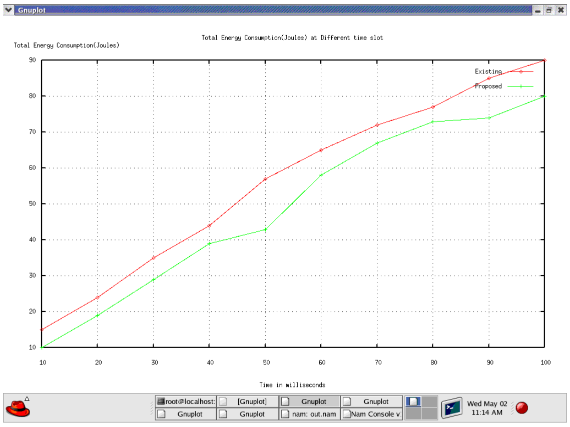


Figure 10. figure showing screenshot for Energy consumption

From figure 9 we can make an observation that the throughput of the proposed system is around 570 Mbps as compared to the throughput of the existing system which is around 460 Mbps.Thus the proposed system scores over the existing system. Similarly for figure 10 the energy consumed by the proposed system is around 80Joules which is less as compared to the existing system having an energy consumption of around 90 Joules.

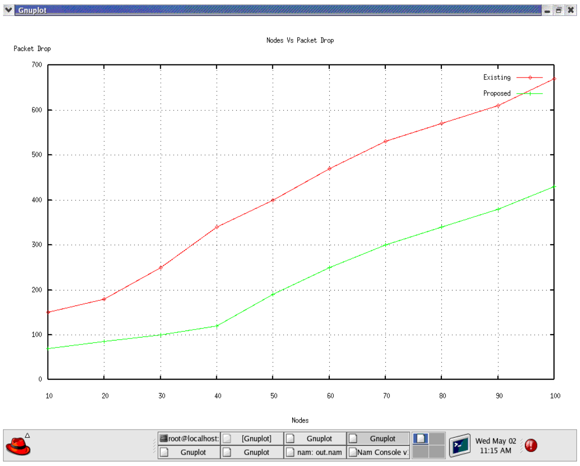


Figure 11. figure showing screenshot for

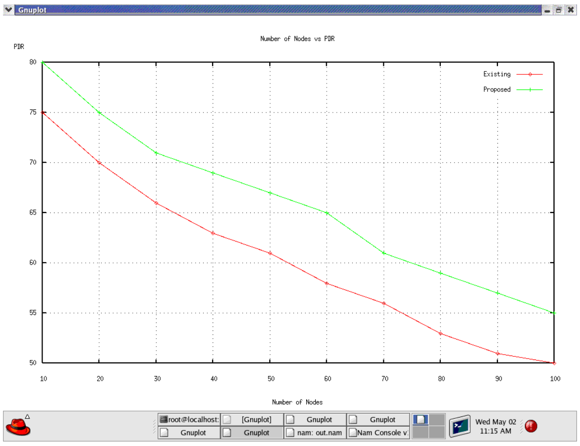


Figure 12. figure showing screenshot for packet packet drop delivery ratio

From figure 5.8 we can make an observation that the packet drop ratio for 100 nodes of the proposed system is around 420 as compared to the packet drop ratio for 100 nodes of the existing system which is around 680. Similarly for figure 5.9 we observe that the packet delivery ratio proposed system is around 80 which is comparatively higher to the packet delivery ratio of the existing system which is around 75.

V. CONCLUSION

We conclude that lots of application are inherently wireless sensor network based wherein they sense the desired parameter and forward the data to the base station to be aggregated and put forward to suitable use as and when desired. In such a case it is always a possibility that a malicious adversary may introduce additional nodes in the network or compromise existing ones. It is therefore highly crucial that high data trustworthiness is high so that correct decisions can be taken. Data provenance becomes critical in evaluating the trustworthiness of sensor data. Low energy and bandwidth consumption, efficient storage and secure transmission are several challenges required to be catered by provenance management .

We also conclude that we have been able to present a scheme to securely transmit provenance for sensor data. The scheme employs in packet bloom filter for providing provenance encoding . The provenance mechanism and reconstruction is taken care by the base station . The provenance scheme has been able to detect packet drop attacks staged by malicious data forwarding nodes.And the proposed scheme scores over the existing schemes in terms of throughput , Energy consumption, the packet drop ratio and the packet delivery ratio.

VI. REFERENCES

[1]. S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure Data Aggregation in Wireless Sensor Networks," IEEE Trans. Information Forensics and Security, vol. 7, no. 3, pp. 1040-1052, June 2012.

[2]. Salmin Sultana, Gabriel Ghinita, Member, IEEE , Elisa Bertino, Fellow, IEEE , and Mohamed Shehab, Member, IEEE Computer Society," A Lightweight Secure Scheme for DetectingProvenance Forgery and Packet DropAttacks in Wireless Sensor Networks", May-june2015.

[3]. Fan Ye,H.Luo,Songwu Lu,Lucia Zhang,"Statistical en-route filtering of injected false data in sensor networks",April 2005.

[4]. Salmin Sultana,Mohamed Shehab,Elisa Bertino,"Secure provenance transmission for streaming data",August 2013.

[5]. S. Madden, J. Franklin, J. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," SIGOPS Operating Systems Review, Dec. 2002.

[6]. H. Lim, Y. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," in Proc. of Data Management for Sensor Networks, pp. 2–7, 2010.

[7]. S. Sultana, E. Bertino, and M. Shehab, "A provenance based mechanism to identify malicious packet dropping adversaries in sensor networks," in Proc. Of ICDCS Workshops, pp. 332–338, 2011.