Asian Journal of Information Technology 15 (20): 3980-3985, 2016
ISSN: 1682-3915
© Medwell Journals, 2016

Accurate Detection of Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks

K.M. Anandkumar and S. Priyadarshini
Department of CSE, Easwari Engineering College, Tamil Nadu, India

Abstract: Wireless sensor networks are used in numerous application domains such as cyber physical infrastructure systems, environmental monitoring etc. Data are produced at a large number of sensor nodes which are deployed in a hostile environment. Data are processed in-network at intermediate hops on their way to a Base Station (BS) to perform decision-making. The data should be trustworthy such that only trustworthy information is considered in the decision process. A malicious adversary may introduce malicious nodes in the network or compromise existing ones and they can modify the data. Data trust worthiness can be assessed by data provenance, since it summarizes the history of ownership and actions performed on the data. Therefore a lightweight secure scheme is used to securely transmit provenance for sensor data. Lightweight secure provenance scheme relies on in-packet Bloom filters to encode provenance. The provenance schema introduces the efficient mechanisms for provenance verification at the base station. Secure provenance scheme is extended with a functionality to detect packet drop attacks staged by malicious nodes.

Key words: Data provenance, security, bloom filter, wireless sensor networks, attacks

INTRODUCTION

Wireless sensor networks are deployed in numerous application domains. Wireless sensor networks combines sensing, computation and communication into a single tiny device. As increasing amounts of valuable information are produced day by day, it is necessary to determine the origin of data. In medicine, science, government and commerce, data provenance tracking is essential for regulatory compliance, rights protection, management of intelligence, medical data and authentication of information as it flows through workplace tasks. Data provenance is an effective way to asses data trustworthiness, since it summarizes the history of ownership and actions performed on the data. By recording provenance we may trace who have contributed to the creation of the data (Hasan *et al*., 2009). We enhance the problem of secure and efficient provenance transmission in the sensor networks and we use provenance scheme to detect packet loss attacks and to identify malicious sensor nodes.

Figure 1 depicts the architecture of WSN. Data provenance allows the BS to trace the source and intermediate nodes of an individual data packet. Our main objective is to securely transmit provenance from source node to destination node. So, we enhance secure provenance scheme to identify malicious nodes and packet dropping attack.

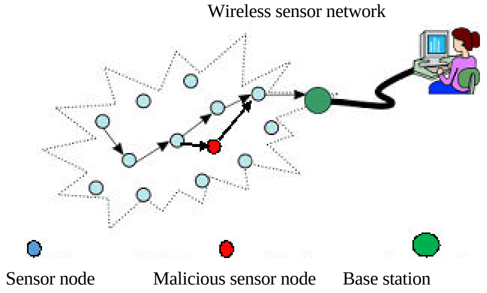

Fig. 1: An Architecture Of WSN with a malicious node

Literature review: Hasan *et al.* (2009) proposed a provenance tracking system which ensures integrity and confidentiality. Provenance tracking system maintain a chain model of provenance. However, this may cause privacy issues, no nodes can really hide their identity since they must put their public key in the records. (Syalim *et al*., 2010) extend this method by applying digital signatures to a directed acyclic graph model of provenance. Ramachandran proposed method called pedigree which captures provenance for network packets in the form of per packet tags that store a history of all nodes that manipulated the packet.

Zhou *et al.* (2010, 2011) proposed a protocol called ExSPAN which describes the history and derivations of network state for distributed systems. It utilize the database notion of data provenance to explain the

Corresponding Author: K.M Anandkumar, Department of CSE, Easwari Engineering College, Tamil Nadu, India

existence of any network state, providing a versatile mechanism for network provenance. However, it does not provides formal security guarantees for provenance data enables new classes of routing algorithms in which decisions can be based not only on the contents of messages but also on the matter in which messages are created and transported.

Bloom Filters are commonly used in networking applications. The attractions of iBF (in-packet Bloom Filters) include credential data path security (Wolf, 2008), IP trace back (Laufer *et al*., 2007), source routing and multicast (Jokela *et al*., 2009) etc. The basic idea in these works is to encode the link identifiers of the routing path into an iBF. However, the encoding of the routing path is performed by the data source, whereas the intermediate routers check their membership in the iBF and forward the packed further based on the decision. However this approach is infeasible for sensor networks where the path may change due to several reasons.

After reviewing the above study, we enhance light weight secure provenance scheme (Sultana *et al*., 2015) which resolves these issues by encoding the provenance in a distributed fashion.

Background and system model: In this study, we discuss about the provenance data and its representation. The threat model and security requirements are presented in this section. Finally, we provide a brief note on bloom filters, their fundamental properties and operations.

MATERIALS AND METHODS

Provenance model: Networks are usually modeled as graphs. We thus model the physical (sensor) network as a graph G (N, L), where the set of nodes, N and the set of links, L, are defined as follows:

- N = {n_i | n_i is a network node of whose identifier is i}
- L = { $l_{i,j}$ | $l_{i,j}$ is an edge connecting nodes n_i and n_j}

Definition 1: An Initiator node generates a data packet and sends it to one or more intermediate node or base station. An intermediate node receives data packet from initiator node or intermediate nodes and it passes them to intermediate nodes or base station. It may also generate an aggregated data packet from the received data packet and send the aggregated packet to intermediate nodes or base station. The base station receives data packet and evaluates the data packet for provenance forgery and packet drop attacks. For a given data packet d, the provenance p_d is represented as a directed acyclic graph of G (V, E), where the set of vertices, V and the set of edges, E, are defined as follows:

- V = {v_i | v_i is a vertex of whose identifier is i}
- E = {$e_{i,j}$ | $e_{i,j}$ is an edge connecting nodes n_i and n_j}

In the provenance scheme, any j^{th} data packet contains) the unique packet sequence number (seq [j])) the previous packet sequence number (pSeq), data value) provenance) generated packet count and received packet count.

Security objectives: We assume that BS is trusted but any other node may be malicious node. An adversary can perform traffic analysis and eavesdrop anywhere on the path. In addition, the adversary is able to deploy a few malicious nodes. The adversary may inject, alter or drop. Packets on the links that are under its control. A data packet with no provenance record will make the data highly suspicious (Hasan *et al*., 2009) and hence generate an alarm at the BS. Our objective is to achieve the following security properties:

Confidentiality: By analyzing the contents of a data packet an unauthorized party cannot gain any knowledge about data provenance. Only BS can process and check the integrity of provenance.

Integrity: Unauthorized party cannot modify provenance information without being detected by the BS.

Freshness: An adversary cannot replay old data and provenance without being detected by the BS. Data-Provenance binding is also important so that an attacker cannot drop or alter the data while retaining the provenance or swap the provenance of two packets.

Bloom filters: In-packet Bloom Filters (BF) (Rothenberg *et al*., 2011) have increasingly become a fundamental data aggregation component to address performance and scalability issues of very diverse network applications, including overlay networks, data-centric routing, traffic monitoring and so on. In this work, focus on the subset of distributed networking applications based on packet-header-size Bloom filters to share some state information set among network nodes. The specific state carried in the bloom filter varies from application to application, ranging from secure credentials to link identifiers with the shared requirement of a fixed-size packet header data structure to efficiently verify set memberships. Considering the constraints faced by the implementation of next generation networks e.g., Gbps speeds, increasingly complex tasks, larger systems, high-speed memory availability, etc., recent inter networking chose to include more information in the

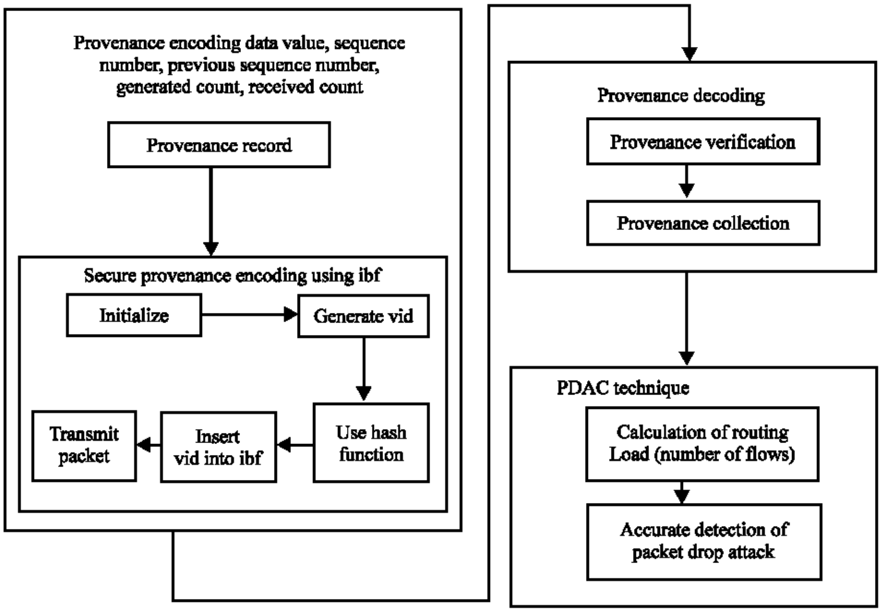
Provenance encoding data value, sequence number, previous sequence number, generated count, received count
Provenance record
Secure provenance encoding using ibf
Initialize
Generate vid
Use hash function
Insert vid into ibf
Transmit packet
Provenance decoding
Provenance verification
Provenance collection
PDAC technique
Calculation of routing Load (number of flows)
Accurate detection of packet drop attack

Fig. 2: Functional architecture

packet headers to keep pace with the increasing speed and needs of Internet-scale systems. The BF used in this type of applications as an in-packet Bloom Filter (iBF). These specific needs may benefit from additional capabilities like element removals or security enhancements.

Detection of provenance forgery: Secure provenance schema securely transmit provenance information from the source node through intermediate nodes to the base station. Encoding provenance information is done at the sender nodes and intermediary nodes. In-Packet Bloom filter (Rothenberg *et al*., 2011) is used to securely transmit provenance information. Each packet consists of a unique sequence number, previous sequence number, data value and an iBF which holds the provenance.

Figure 2 depicts the functional architecture. In this study, we focus on securely transmitting provenance information to the BS. In an aggregation infrastructure securing the data values is an important aspect but that has been already addressed in previous research work provenance scheme can be used in conjunction with previous works to obtain a solution that provides security for provenance, data and data-provenance binding.

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 (vertex ID) is generated per-packet based on the packet sequence number (seq). We use polynomial hash function to produce VID in a secure manner. Thus for a given data packet, the VID of a vertex representing the node n_i is computed as:

$$\text{Vidi} = \text{generate VID (ni, seq [j])} \tag{1}$$

Equation 1 represents vid_i generation for a sensor node n_i. When a source node generates a packet, it also creates a BF (referred to as ibf_0), initialized to 0. The source then generates a vertex according to Eq. 1, inserts the VID into ibf_0 and transmits the BF as a part of the packet. n_i updates the provenance of the packet by inserting vid_i into the iBF.

Provenance decoding at the base station: When the BS receives the data packet, it executes the provenance verification process and provenance collection process. Provenance collection process is done at the BS only if the provenance verification process didn't succeed.

Provenance verification: The BS conducts the provenance verification process to verify its knowledge of provenance and to check the integrity of the transmitted provenance.

ALGORITHM 1: Provenance verification

```
INPUT: Received Packet with sequence $seq_i$ and ibf. Set of hash functions H, Data path P
$Bf_c$ = 0//initialize bloom filter
for each $n_i$ _ P do
    $Vid_i$ = generate VID ($n_i$, seq)
    Insert $vid_i$ into $BF_c$ using hash function H
End for
If ($BF_c$= ibf)
    Return true//Provenance is verified
End if
Return false
```

Provenance verification: Figure 3 represents the workflow of provenance verification process. At first, BS initializes a bloom filter BFc with all 0's. BF_c is the bloom filter constructed in the base station. The BF_c is then updated by generating the VID for each node in the path P and inserting this VID into the BF. BFc is the bloom filter constructed by the base station and ibf is the received bloom filter with the packet. Now, BFc and ibf are compared to check whether they are equal. Verification failure here indicates either a change in the data flow path, a packet drop attack or BF modification attack and triggers the provenance collection process.

Provenance collection: The provenance collection scheme makes a list of potential vertices in the provenance graph through the ibf membership testing over all the nodes.

Algorithm 2: Provenance collection

```
INPUT: Received packet with sequence seq, previous sequence and IBF ibf
Set of hash functions H, Set of nodes in the (N) network.
1. Initialize
Set of possible nodes S←0
Bloom Filter $BF_c$←0
2. Determine possible nodes in the path and build Representative BF
For each ni'€ P' do
    vidi = generate VID (ni, seq)
    if (vid'i is in ibf) then
    S←S U ni
   insert vid'i into BFc using hash functionsH
end if
end for
3. Verify BFc with the received iBF
 if (BFc= ibf)
     Return S//Provenance is verified
else
      Return NULL//Indicates an in-transit attack
end if
```

Provenance collection: Figure 4 represents the workflow of provenance collection process. For each node n_i in the network, the BS creates the corresponding vertex (i.e., v_i with VID vid_i) using Eq. (1). The BS then performs the membership query of vid_i within ibf. If the algorithm returns true, the vertex is very likely present in the provenance, i.e., the host node n_i is in the data path. Once the BS finalizes the set of potential candidate nodes, it executes the verification algorithm on this set. On verifying BF_c with the received ibf, if it succeeds we decide that it was a natural path change. Otherwise, an attack has occurred.

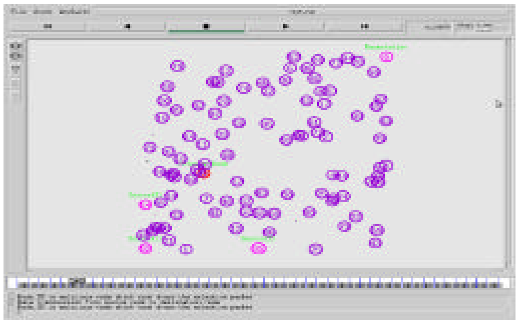

Fig. 3: Data transmission from source node to base station with a malicious node

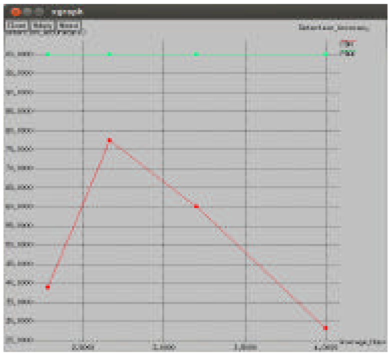

Fig. 4: Comparative graph for attack detection accuracy

RESULTS AND DISCUSSION

Detection of packet drop attacks: The secure provenance encoding schema is extended to detect packet drop attack and to identify malicious nodes. We consider only linear data flow paths. For a data packet, along with the provenance record generated by a node, data packet will contain an acknowledgement in the form of a previous sequence number ($pseq_i$) of the lastly seen packet belonging to the data path. Lastly seen packet can be a processed packet or a forwarded packet. If there is an intermediate packet drop, some nodes on the path do not receive the packet. Hence, during the next round of packet transmission, there will be a mismatch between the acknowledgements generated from different nodes on the path. We utilize this fact to identify the malicious node and to detect packet drop attacks.

Packet dropping in the network may due to several reasons in the network. Existing provenance scheme does not classify the packet dropping that arises due to congestion in the path which leads to false negative. In case of a malicious router selectively dropping packets, it is very difficult to term missing of packets to malicious behavior as normal congestion in today's network can bring about similar behavior at the routers. Routers drop packets when the inflow of packets exceeds their buffering capacity. Hence a technique known as PDAC (Packet Drop due to Attack or Congestion) is contributed which classifies packet dropping due to congestion or attack and detects malicious node accurately. BS collects the routing load on every router in the path and classifies packet loss reason. Routing load refers to the number of data flows for a particular node. If it involves in more number of flows then probability packet dropping at the node is high. This technique classifies dropping reason accurately and improves attack detection accuracy. PDAC technique is used in conjunction with the previous research work to accurately detect packet drop attacks and provenance forgery.

Data packet representation: The packet header must securely propagate the packet sequence number generated in the previous round. Thus in Thus, in the extended provenance scheme, any jth data packet contains) the unique packet sequence number (seq [j])) the previous packet sequence number (pSeq) a data value) Provenance and Generated packet count and received packet count.

Suspicious nodes detection: Every node in the data path, generates/receives data packet j which contains unique packet sequence number seq [j] and a previous sequence number pSeq$_i$. A node must contain a per-flow record to store the previous packet sequence number for each data packet that passed through the node. If a node receives a packet from a data flow for which it has no previous packet information, then it may use a pre-specified special purpose identifier such as 0 as the previous packet sequence pSeq$_i$.

At the base station, for a received data packet j the difference between the current sequence number seq [j] of the received packet and lastly seen sequence number p Seq by the base station is taken to generate a suspicious nodes list. Based on the difference between the sequence numbers, the adjacent link to the suspicious node is identified. If the difference is >1 then, it is added to suspicious list. Only with the difference between the sequence numbers we cannot conclude that the node is a malicious node, because it might have dropped the packet due to congestion or there can be a delay in sending a data packet.

PDAC (Packet Drop due to Attack or Congestion) technique: Once the suspicious node list is generated, the difference between the generated packet count and received packet count is taken for a suspicious node. Difference found is the packet loss count. The difference is compared with the static threshold. The static threshold is a user defined threshold. If the determined packet losses is less than the threshold value, then the node is not compromised and it should be due to congestion. Too many losses would mean malicious intent. In order to avoid false positives, the threshold must be large enough. If the determined loss is higher than the threshold, we conclude that that the packet loss was due to an attack.

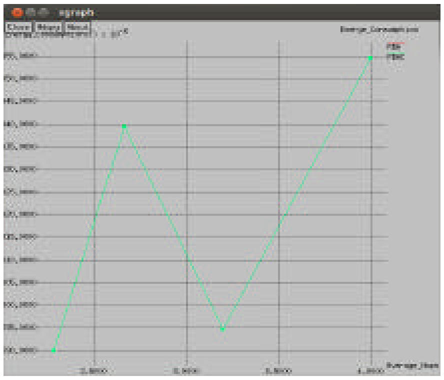

Fig. 5: Xgraph for energy consumption

Implementation and analysis: The provenance encoding schema has been implemented using network simulator NS2. The proposed schema utilizes network topology compromising of 100 wireless nodes. The mobility model used is Two Ray Ground with simulation time 100s and scenario 600×600m. Communication range is varied for analysis as 200, 250, 300 and 350m to vary the hops. There is a significant change in the attack detection accuracy.

Figure 3 shows the data transmission from source node to destination node. As discussed above, base station initializes the ibf and encodes vid into the ibf. The provenance record is carried by the bloom filter and processed in-network at intermediate nodes on their way to the base station. We use AODV (Ad-Hoc on demand distance vector routing protocol) for implementing the secure provenance encoding and decoding schema.

Figure 4 depicts the xgraph between Packet Drop Attack Detection (PDA) scheme and Packet drop due Attack or Congestion technique. Selective packet dropping attack is considered in this process. Analysis is undergone with a Malicious node is inserted in the network which performs selective packet dropping attack. With the variation of number of hops the malicious detection accuracy is better in PDAC technique when compared to existing packet drop attack detection scheme. Figure 5 depicts the xgraph for energy consumption. With variation of number of hops the energy consumption is similar in both packet dropping attack scheme and packet drop due to attack or congestion scheme. So, the enhanced scheme is also a lightweight secure scheme.

CONCLUSION

The secure provenance schema focuses on accurate detection of provenance forgery and packet loss attacks. The problem of securely transmitting provenance for sensor networks has been solved by proposing a secure provenance encoding and decoding scheme based on bloom filters. The scheme ensures confidentiality, integrity and freshness of provenance. We consider only selective packet dropping attacks. In future we would like to do more experiments, considering different kind of attacks in wireless sensor network.

REFERENCES

Hasan, R., R. Sion and M. Winslett, 2009. The case of the fake picasso: preventing history forgery with secure provenance. FAST., 9: 1-14.

Jokela, P., A. Zahemszky, E.C. Rothenberg, S. Arianfar and P. Nikander, 2009. LIPSIN: Line speed publish subscribe inter-networking. ACM. SIGCOMM. Comput. Commun. Rev., 39: 195-206.

Laufer, R.P., P.B. Velloso, D.D.O. Cunha, I.M. Moraes and M.D. Bicudo *et al*., 2007. Towards stateless single-packet IP traceback. Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN 2007), October 15-18, 2007, IEEE, Dublin, Ireland, ISBN: 0-7695-3000-1, pp: 548-555.

Rothenberg, C.E., C.A.B. Macapuna, M.F. Magalhaes, F.L. Verdi and A. Wiesmaier, 2011. In-packet bloom filters: Design and networking applications. Comput. Netw., 55: 1364-1378.

Sultana, S., G. Ghinita, E. Bertino and M. Shehab, 2015. A lightweight secure scheme for detecting provenance forgery and packet dropattacks in wireless sensor networks. IEEE. Trans. Dependable Secure Comput., 12: 256-269.

Syalim, A., T. Nishide and K. Sakurai, 2010. Preserving Integrity and Confidentiality of a Directed Acyclic Graph Model of Provenance. In: Data and Applications Security and Privacy. Sara, F. and S. Jajodia (Eds.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-642-13738-9, pp: 311-318.

Wolf, T., 2008. Data path credentials for high-performance capabilities-based networks. Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, November 6-7, 2008, ACM, San Jose, California, USA., ISBN: 978-1-60558-346-4, pp: 129-130.

Zhou, W., M. Sherr, T. Tao, X. Li and B.T. Loo *et al*., 2010. Efficient querying and maintenance of network provenance at internet-scale. Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, June 6-11, 2010, ACM, Indianapolis, Indiana, USA., ISBN: 978-1-4503-0032-2, pp: 615-626.

Zhou, W., Q. Fei, A. Narayan, A. Haeberlen and B.T. Loo *et al*., 2011. Secure network provenance. Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, October 23-26, 2011, ACM, Cascais, Portugal, ISBN: 978-1-4503-0977-6, pp: 295-310.