International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Special Issue-3, May 2016 E-ISSN: 2347-2693

Sspp: Secure Scheme for Provenance Forgery and Packet Drop Detection in Wireless Sensor Network

Ujwala N[1], Siddaramappa V[2]

[1,2] *Department of CSE, City Engineering College, India*

Available online at: www.ijcseonline.org

Abstract— Deployment of sensor networks are used in many applications data that are collected by these sensors are used in decision making such as in Supervisory control and data Acquisition, Battlefield monitoring systems and in many other applications. Data are collected from multiple sources through intermediate nodes. Malicious adversary may introduce additional nodes in the network or compromise existing ones. So trustworthiness of the data collected from different sources is important for decision making. In evaluating the trustworthiness of the sensor data provenance plays a key factor. Several challenges that affect the provenance management include secure transmission and efficient storage, bandwidth consumption and low energy. A novel lightweight schema was proposed to transmit provenance of the sensor data securely. Bloom filters are used in the proposed technique to encode provenance. Efficient mechanisms are introduced for reconstruction of the base station and provenance verification. The secure provenance scheme was extended with the functionality to detect packet drop attacks done by malicious data forwarding nodes. We evaluate the proposed technique and the results prove the effectiveness and efficiency of the lightweight secure provenance scheme in detecting packet forgery and loss attacks.

Keywords—*Trustworthiness, Bloom filters, wireless sensor network.*

I. 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 innetwork at intermediate hops on their way to a base station (BS) that performs decision-making. 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. 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. We 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. As opposed to existing research that employs separate transmission channels for data and provenance [4], we only require a single channel for both. Furthermore, traditional provenance security solutions use intensively cryptography and digital signatures [5], and they employ append-based data structures to store provenance, leading to prohibitive costs.

Our specific contributions are:

- We formulate the problem of secure provenance transmission in sensor networks.
- We propose an in-packet Bloom filter (iBF) provenance-encoding scheme.
- We design efficient techniques for provenance decoding and verification at the base station.
- We extend the secure provenance encoding scheme and devise a mechanism that detects packet drop attacks staged by malicious forwarding sensor nodes.

II. RELATED WORK

Family [26] catches provenance for system bundles according to parcel labels that store a past filled with all hubs and procedures that controlled the parcel. Be that as it may, the plan accepts a trusted domain which is not realistic in sensor systems.

ExSPAN [27] depicts the history and inferences of system express that outcome from the execution of a disseminated convention. This framework additionally does not address security concerns and is particular to some system use cases. SNP stretches out system provenance to adversarial situations. Since these frameworks are broadly useful system provenance frameworks, they are not optimized for the asset obliged sensor systems. To develop quickly,

transmission of a lot of demonstrate provenance data alongside information will bring about critical transfer speed overhead, consequently low productivity and adaptability.

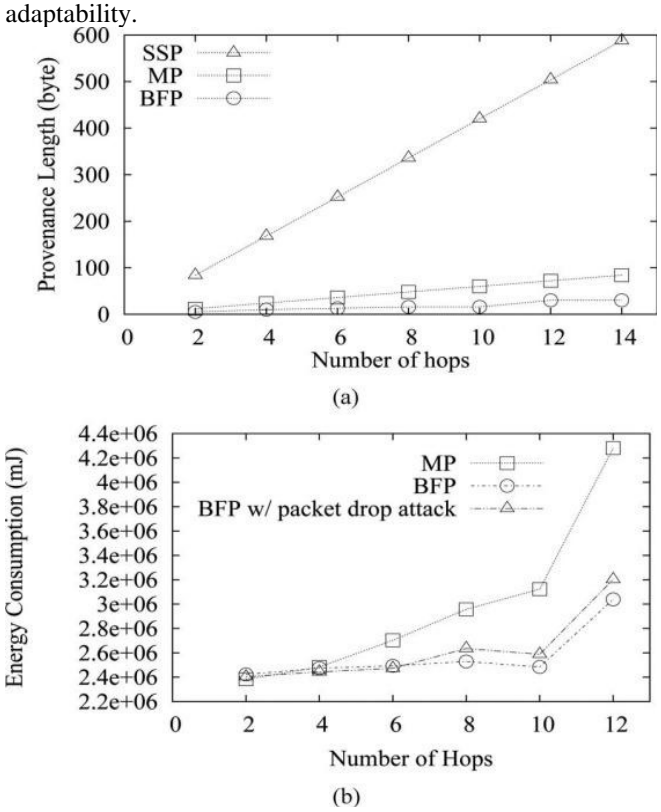

Fig (a) Provenance length (b) Energy consumption

Vijayakumar propose an application particular framework for close ongoing provenance accumulation in information streams. In any case, this framework follows the wellspring of a stream long after the procedure has finished.

III. ARCHITECTURE

We propose a distributed mechanism to encode provenance at the nodes and a centralized algorithm to decode it at the BS. The technical core of our proposal is the notion of inpacket Bloom filter [11]. Each packet consists of a unique sequence number, data value, and an Ibf which holds the provenance.

A. 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 k_i of the host node.

We use 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 is computed as

$$vi\ di = generateVID\ ni\ ,\ seq = Eki(seq\ ) \ldots .(1)$$

where E is a secure block cipher such as AES, etc.

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 Eq. (1), inserts the VID into ibf0 and transmits the BF as a part of the packet.

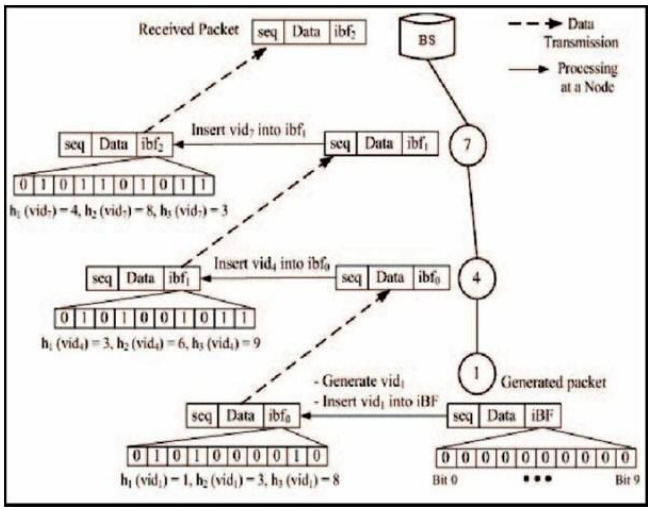

Fig 1: encoding provenance

We use only fast message authentication code (MAC) method and Bloom filter, which are fixed size data structures that represent provenance. Bloom filters make best usage of bandwidth, and they yield low error rates in practice. We formulate the problem of secure provenance transmission in wireless sensor networks, and identify the challenges specific to this context. We propose an iBF (in packet Bloom filter) provenance encoding mechanism also design efficient techniques for provenance decoding and verification at the base station. We extend the secure provenance encoding mechanism and devise a mechanism that detects data packet drop attacks step by malicious forwarding sensor nodes.

B. Provenance Decoding

When the BS receives a data packet, it executes the provenance verification process, which assumes that the BS knows what the data path should be, and checks the iBF to see whether the correct path has been followed. However, right after network deployment, as well as when the topology changes (e.g., due to node failure), the path of a packet sent by a source may not be known to the BS. In this case, a provenance collection process is necessary, which retrieves provenance from the received iBF and thus the BS learns the data path from a source node.

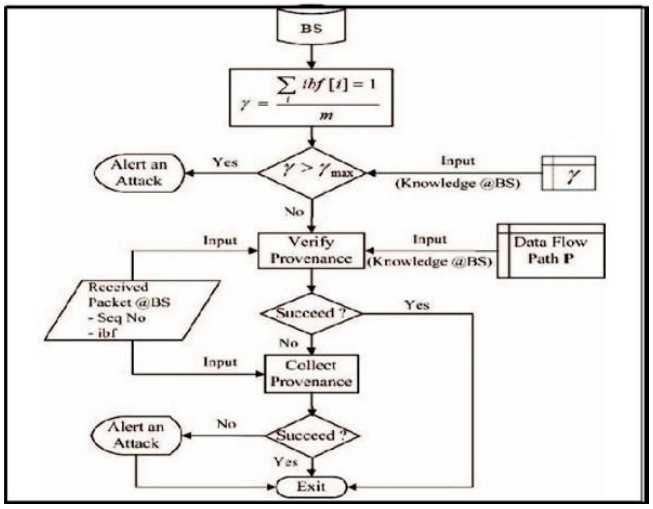

Fig 2: Provenance processing

Provenance verification: The BS conducts the verification process not only to verify its knowledge of provenance but also to check the integrity of the transmitted provenance. Algorithm 1 shows the steps to verify provenance for a given packet. We assume that the knowledge of the BS about this packet's path is P0.

Algorithm 1: Provenance Verification
STEP 1: Input: Received packet with sequence seq and iBF *ibf*. Set of hash functions H, Data path $P' =< nl1', \ldots n1', \ldots, np' >$
STEP 2: $BFc \leftarrow 0$ // Initialize Bloom Filter
STEP 3: for each n $i \in P'$ **do**
STEP 4: $vidi'$ = generate VID (ni', seq)
STEP 5: Insert $vidi'$ into BFc using hash functions in H
STEP 6: endfor
STEP 7: if ($BFc = ibf$) **then**
STEP 8: return true // Provenance is verified
STEP 9: endif
STEP 10: return false

Provenance collection: As illustrated in Algorithm 2, 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) the false positive probability obtained is very low.
Algorithm 2: Provenance Collection
Input: Received packet with sequence *seq* and iBF *ibf*.
Set of nodes (N) in the network, Set of hash functions H
1. Initialize
Set of Possible Nodes S ← ∅
Bloom Filter $BF_c \leftarrow 0$ // To represent S
2. Determine possible nodes in the path and build the representative BF
for each node $n_i \in$ N **do**
vid_i = generateVID (n_i, seq)
if (vid_i is in ibf) **then**
S ← S ∪ n_i
Insert vid_i into BF c using hash functions in H
endif
endfor
3. Verify BF_c with the received iBF
if ($BF_c = ibf$) **then**
return S // Provenance has been determined correctly
else
return NULL // Indicates an in-transit attack
endif

C. Provenance Data binding

One of the important security challenges for a provenance scheme is to tie-up data and provenance. In an aggregation infrastructure, the data value is updated at each intermediate node which makes it a crucial problem to maintain the relationship between provenance and the intermediate data. A trivial solution can be based on making the provenance encoding mechanism dependent on the partial aggregation results (PAR) and append each PAR to the packet to verify the data-provenance binding at the BS coupling is ensured at each node in the routing path.

IV. DETECTION OF PACKET DROP ATTACKS

We extend the secure provenance encoding scheme to detect packet drop attacks and to identify malicious node (s). We assume the links on the path exhibit natural packet loss and several adversarial nodes may exist on the path. For simplicity, we consider only linear data flow paths. Also, we do not address the issue of recovery once a malicious node is detected. Existing techniques that are orthogonal to our detection scheme can be used, which may initiate multipath routing or build a dissemination tree around the compromised nodes.

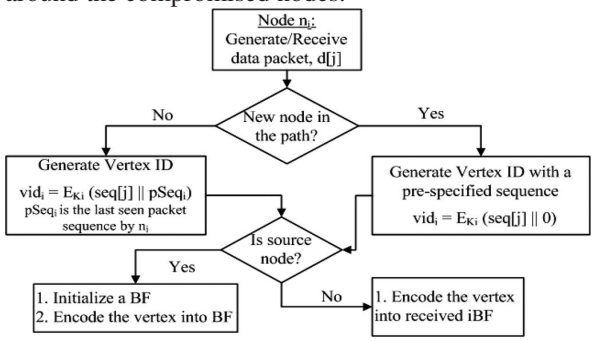

Fig 3: Extended provenance Framework

acknowledgements generated from different nodes on the path. We utilize this fact to detect the packet drop attack and to localize the malicious node. We describe next packet representation, provenance encoding and decoding for detecting packet loss.

A. Representation of data packet

To enable packet loss detection, a packet header must securely propagate the packet sequence number generated by the data source in the previous round. In addition, as in the basic scheme, the packet must be marked with a unique sequence number to facilitate per-packet provenance generation and verification. Thus, in the extended provenance scheme, any jt data packet contains 1) the unique packet sequence number (*seq½j*), 2) the previous packet sequence number (*pSeq*), 3) a data value, and 4) provenance.

B. Provenance encoding

The provenance record of a node includes 1) the node ID, and 2) an acknowledgement of the lastly observed packet in the flow. The acknowledgement can be generated in various ways to serve this purpose. In our solution, a node ni creates a vertex vi for every jth packet it generates/forwards. The vertex ID *vidi* is generated as *vidi* ¼ generate VID (3) where *pSeqi* is the knowledge of ni about the sequence number of the previous packet in the flow. ni updates the provenance of the packet by inserting *vidi* into the iBF.

C. Provenance decoding at 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.

Provenance verification

The BS knows 1) the current data path for the packet (decoded from the provenance of the 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.

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.

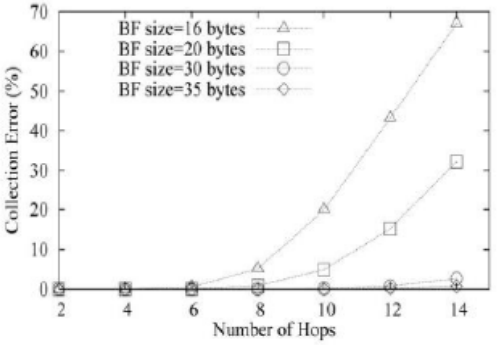

Fig 4: Collection error

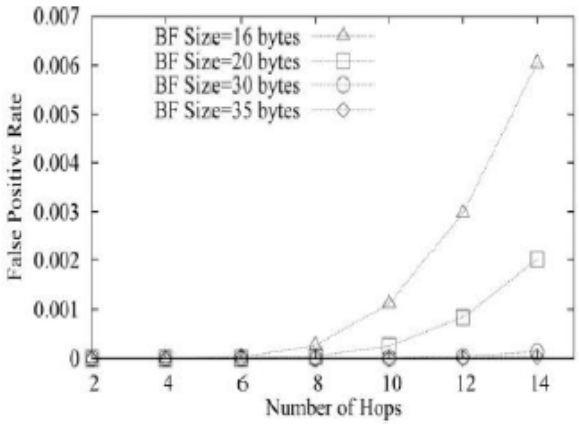

Fig 5: False positive rate

CONCLUSION

We tended to the issue of safely transmitting demonstrate provenance for sensor organizes, and proposed a light-weight provenance encoding and deciphering plan in view of Bloom channels. The plan guarantees privacy, uprightness and freshness of provenance. We extended the plan to consolidate information provenance tying, and to incorporate parcel succession data that backings discovery of bundle misfortune assaults. Exploratory and explanatory assessment results demonstrate that the proposed plan is successful, light-weight and versatile. In future work, we plan to actualize a genuine framework model of our safe provenance conspire, and to enhance the exactness of parcel misfortune location, particularly on account of numerous sequential noxious sensor hubs.

REFERENCES

[1] H. Lim, Y. Moon and E. Bertino, "Provenance based trustworthiness assessment in sensor network" proc seventh int'l workshop, pp. 2-7, 2010.

[2] I. Foster, J. Vockler, "Chimera: A virtual data system for representing, querying and automating

data derivation" proc. conf. scientistical database management, pp. 37-46, 2002.
[3] K. Muniswamy-Reddy, D. Holland, "Provenance-A ware storage systems" proc. USENIX Ann. Technical conf., pp. 4-4, 2006.
[4] Y. Simmhan, B. Plale, "A survey of data provenance in E-Science" ACM SIGMOD record, vol. 34, pp. 31-36, 2005.
[5] Rohit Aggarwal and Khushboo Bansal , "An Efficient Intruder Detection System against Sinkhole Attack in Wireless Sensor Networks: A Review", International Journal of Computer Sciences and Engineering, Volume-04, Issue-04, Page No (64-68), Apr -2016