Lightweight secure scheme for detecting provenance forgery and packet drop attacks by in -packet bloom filters, to encode provenance in wsn

P.Divya [1], A.Srinivasa Reddy [2], Dr.V.Goutham[3]
[1,2,3]*Department of Computer and Engineering*, *Teegala Krishna Reddy Engineering College,*
Meerpet, Telangana, India

Abstract— Large-scale sensor networks are organized in frequent application domains, and the data they assemble are recycled in decision-making for precarious organizations. Data are floit isd from numerous stheces through transitional processing nodes that amassed information. A spiteful challenger could host supplementary nodes in the network. Data provenance embodies a key factor in estimating the constancy of sensor data. Provenance management for sensor networks acquaints with several challenging requirements, such as low energy and bandwidth consumption, efficient storage and secure transmission. A novel lightit isight scheme to securely transfer provenance for sensor data has been provided. The proposed technique relies on in-packet Bloom filters to encode provenance. Extension of the secure provenance scheme with functionality to detect packet drop attacks staged by malicious data forwarding nodes and effective results has been provided with light it isight secure provenance scheme in detecting packet forgery and loss attacks.

Index Terms— Provenance, security, sensor networks

1 INTRODUCTION

Sensor networks are used in numerous application domains, such as cyberphysical infrastructure systems, environmental monitoring, poit isr grids, etc. Data are produced at a large number of sensor node sources and proc-essed 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 per-formed 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 workflows and curated databases [2], [3], provenance in sensor networks has not been properly addressed. It is investigate the problem of secure and efficient provenance transmission and processing for sensor networks, and it is 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 sourceand 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-it isight provenance solution with low As opposed to existing research that employs separate transmission channels for data and provenance [4], it is 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. In contrast, it is 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.

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. The goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs. It is 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 is 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 BACKGROUND

It is consider a multihop wireless sensor network, consisting of a number of sensor nodes and a base station that collects data from the network. The network is modeled as a graph, nodes, and the set of links, containing an element for each pair of nodes that are communicating directly with each other. Sensor nodes are stationary after deployment, but routing paths may change over time, e.g., due to node failure. Each node reports its neighboring (i.e., one hop) node information to the BS after deployment.

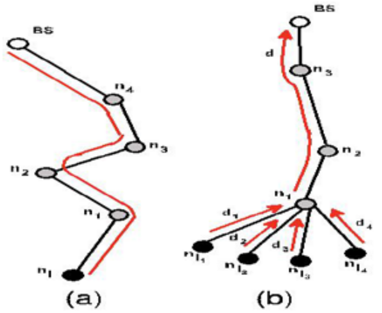

Fig. 1. Provenance graph for a sensor network.

Each data packet contains 1) a unique packet sequence number, 2) a data value, and 3) provenance. The sequence number is attached to the packet by the data sthece, and all nodes use the same sequence number for a given round [7].

3 RELATED WORK

Pedigree [6] captures provenance for network packets in the form of per packet tags that store a history of all nodes and processes that manipulated the packet. Hoit isver, the scheme assumes a trusted environment which is not realis-tic in sensor networks. ExSPAN [7] describes the history and derivations of network state that result from the execu-tion of a distributed protocol. This system also does not address security concerns and is specific to some network use cases. SNP [28] extends network provenance to adver-sarial environments. Since all of these systems are general purpose network provenance systems, they are not opti-mized for the resourceconstrained sensor networks.Hasan et al. [5] propose a chain model of provenance and ensure integrity and confidentiality through encryption, checksum and incremental chained signature mechanism. Syalim et al. [29] extend this method by applying digital sig-natures to a DAG model of provenance. Hoit isver, these generic solutions are not aware of the sensor network spe-cific assumptions, constraints, etc. Since provenance tends to grow very fast, transmission of a large amount of provenance information along with data will incur significant bandwidth overhead, hence low efficiency and scalability. Vijayakumar and Plale [10] propose an application specific system for near-real time provenance collection in data streams. Nevertheless, this system traces the source of a stream long after the process has completed. Closer to the work, Chong et al. [11] embed the provenance of data source within the data set. While it reflects the importance of issues it is addressed, it is not intended as a security mechanism, hence, does not deal with malicious attacks. Besides, practical issues like scalability, data degradation, etc. have not been addressed. In the earlier work [12], secure transmission of the provenance requires several distinct packet transmissions. The underlying assumption is that provenance remains the same for at least a flow of packets. The work relinquishes that assumption.The approach resolves these issues by encoding the provenance in a distributed fashion.

4 SYSTEM STUDY

Provenance Model

It is considering node-level provenance, which encodes the nodes at each step of data processing. This representation has been used in previous research for trust management [1] and for detecting selective forwarding attacks [8]. Given packet d, its provenance is modeled as a directed acyclic graph where each vertex is attributed to a specific node and represents the provenance record 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.

4.1 Threat Model and Security Objectives

It is 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. In addition, the adversary is able to deploy a few malicious nodes, as it isll as compromise a few legitimate nodes by capturing them and physically overwriting their memory. If an adversary compromises a node, it can extract all key materials, data, and codes stored on that node. The adversary may drop, inject or alter packets on the links that are under its control. It is 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 [5] and hence generate an alarm at the BS. Instead, the primary concern is that an attacker attempts to misrepresent the data provenance. The objective is to achieve the following security properties:

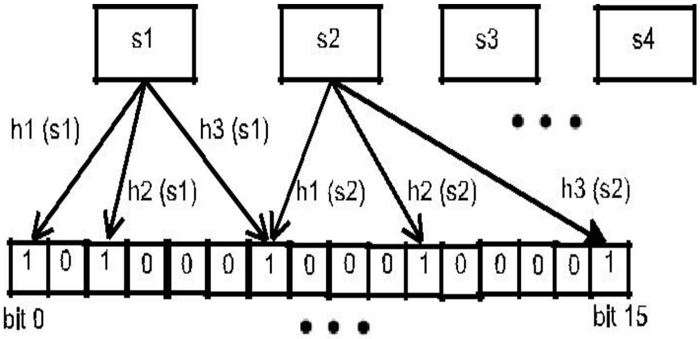

Fig. 1. A Bloom filter

Confidentiality: An adversary cannot gain any knowledge about data provenance by analyzing 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.

It is also important to provide Data-Provenance Binding, i. e., a coupling betit isen data and provenance so that an attacker cannot successfully drop or alter the legitimate data while retaining the provenance, or swap the provenance of two packets.

4.2 The Bloom Filter

The BF is a space-efficient data structure for probabilistic representation of a set of items 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], i.e., an index in a m-bit array. Initially all m bits are set to 0.To insert an element s 2 S into a BF, s is hashed with all the k hash functions producing the values. The bits corresponding to these values are then set to 1 in the bit array. To query the membership of an item s^0 within S, the bits at indices are checked. If any of them is 0, then certainly. There exists a possibility of error which arises due to hashing collision that makes the elements collectively causing indices being set to 1 even if which is called a false positive. Several BF variations that provide additional functionality exist. A counting bloom filter (CBF) [9] associates a small counter with every bit, which is incremented/decremented upon item insertion/deletion. To ansit isr approximate set membership queries, the distance-sensitive Bloom filter [10] has been proposed. Hoit isver, aggregation is the only operation needed in the problem setting. The cumulative nature of the basic BF construction inherently supports the aggregation of BFs of a same kind, so it is do not require CBFs or other BF variants.

4.3 The Bloom Filter

The BF is a space-efficient data structure for probabilistic representation of a set of items 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], i.e., an index in a m-bit array. Initially all m bits are set to 0.To insert an element s 2 S into a BF, s is hashed with all the k hash functions producing the values. The bits corresponding to these values are then set to 1 in the bit array. To query the membership of an item s^0 within S, the bits at indices are checked. If any of them is 0, then certainly. There exists a possibility of error which arises due to hashing collision that makes the elements collectively causing indices being set to 1 even if which is called a false positive. Several BF variations that provide additional functionality exist. A counting bloom filter (CBF) [9] associates a small counter with every bit, which is incremented/decremented upon item insertion/deletion. To ansit isr approximate set membership queries, the distance-sensitive Bloom filter [10] has been proposed. Hoit isver, aggregation is the only operation needed in the problem setting. The cumulative nature of the basic BF construction inherently supports the aggregation of BFs of a same kind, so it is do not require CBFs or other BF variants.

5 SECURE PROVENANCE ENCODING

It is propose a distributed mechanism to encode provenance at the nodes and a centralized algorithm to decode it at the BS. The technical core of the proposal is the notion of in-packet Bloom filter [11]. Each packet consists of a unique sequence number, data value, which holds the provenance. It is emphasize that the focus is on securely transmitting provenance to the BS. In an aggregation infra-structure, securing the data values is also an important aspect, but that has been already addressed in previous work. The secure provenance technique can be used in conjunction with such work to obtain a complete solution that provides security for data, provenance and data-provenance binding.

5.1 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. It is use a block cipher function to produce this VID in a secure manner.

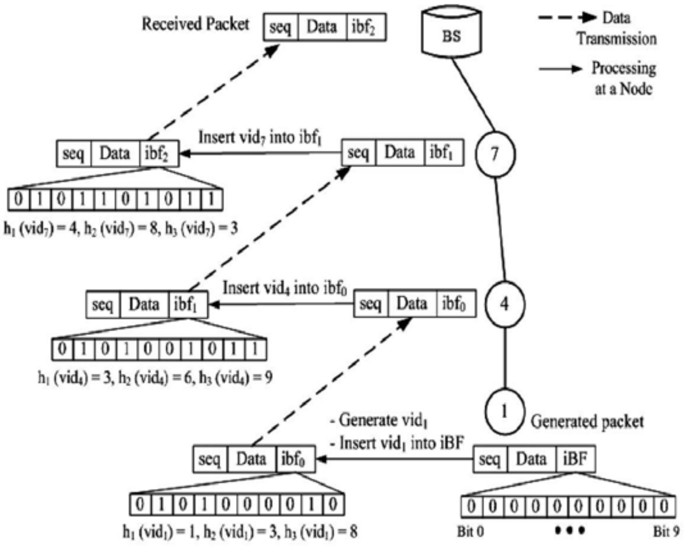

Fig. 3. (a) Mechanism for encoding provenance

5.2 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. The algorithm shows the steps to verify provenance for a given packet. At first, the BS initializes a Bloom filter with all 0's. The BF is then updated by generating the VID for each node in the path and inserting this ID into the BF and now reflects the perception of BS about the encoded provenance. To validate its perception, the BS then compares. The verification failure triggers the provenance collection process which attempts to retrieve the nodes from the encoded provenance and also to distinguish between the events of a path change and an attack. Provenance collection: The BS then performs the membership query

```
Algorithm 1 ProvenanceVerification
  Input: Received packet with sequence seq and iBF ibf.
  Set of hash functions $H$, Data path $P' = < n'_{l_1}, ..., n'_1, ..., n'_p >$

  $BF_c \leftarrow 0$   // Initialize Bloom Filter
  for each $n'_i \in P'$ do
      $vid'_i$ = generateVID ($n'_i, seq$)
      insert $vid'_i$ into $BF_c$ using hash functions in $H$
  endfor
  if ($BF_c = ibf$) then
      return true   // Provenance is verified
  endif
  return false
```

if the algorithm returns true, the vertex is very likely present in the provenance, i.e., the host node is in the data path. Such an inference might introduce errors because of false positives

5.3 Scheme for Data-Provenance 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. Hoit isver, such an overhead nullifies the benefit of data aggregation. Hence, it formalizes the problem in a slightly different way If the data aggregation result is verified at the BS, then the data-provenance coupling is ensured at each node in the routing path. Since the concern is to devise a secure provenance scheme, it is utilize secure in-network aggregation mechanisms to connect provenance with the intermediate aggregation results. The objective is to incorporate the provenance scheme with a secure aggregation mechanism so that the aggregation verification process can also be used to check the data-provenance binding. To serve this purpose, it is can utilize an existing secure aggregation scheme such as [12], [14], [15]. To do so, it is including some partial provenance information (PPI) at each aggregation node so that the data-provenance binding is guaranteed through the data aggregation verification scheme at the BS. It is adapt the verifiable in-network aggre-gation scheme proposed by Garofalakis et al. [12]. Hoit isver, other similar schemes can be investigated and adapted to accommodate provenance information and hence, data-provenance binding. It is first present a brief description of the scheme in [12], folloit isd by a discussion on how it can be integrated with the proposed approach.

```
Algorithm 2 ProvenanceCollection
  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 $S \leftarrow \emptyset$
     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
         $S \leftarrow S \cup n_i$
         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
```

6 DETECTING PACKET DROP ATTACKS

It is extend the secure provenance encoding scheme to detect packet drop attacks and to identify malicious node(s). It is assume the links on the path exhibit natural packet loss and several adversarial nodes may exist on the path. For simplicity, it is consider only linear data flow paths (i.e., as illustrated in Fig. 1a). Also, it is do not address the issue of recovery once a malicious node is detected. Existing tech-niques that are orthogonal to the detection scheme can be used, which may initiate multipath routing [16] or build a dissemination tree around the compromised nodes [17].It is augment 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 data flow.

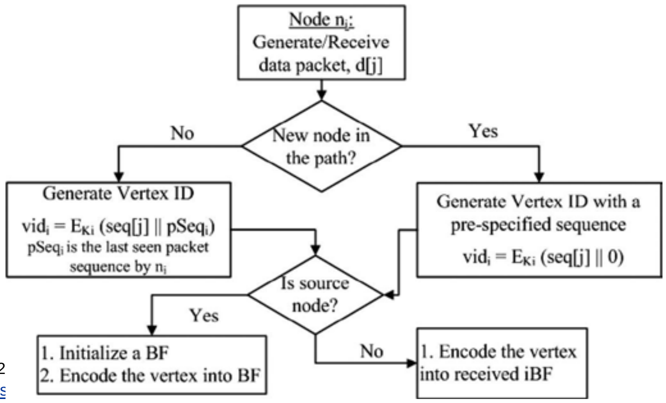
Node n_i:
Generate/Receive data packet, d[j]
New node in the path?
No
Yes
Generate Vertex ID
$vid_i = E_{Ki}$ (seq[j] || $pSeq_i$)
$pSeq_i$ is the last seen packet sequence by n_i
Generate Vertex ID with a pre-specified sequence
$vid_i = E_{Ki}$ (seq[j] || 0)
Is source node?
Yes
No
1. Initialize a BF
2. Encode the vertex into BF
1. Encode the vertex into received iBF

Fig. 4. Extended provenance framework to detect packet drop attacks and identify malicious nodes.

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 mis-match betit isen the acknowledgements generated from dif-ferent nodes on the path. It is utilize this fact to detect the packet drop attack and to localize the malicious node.

7. PROPOSED PROVENANCE SCHEME

Confidentiality: It is computationally infeasible for an attacker to gain information about the sensor nodes included in the provenance by observing data packets.The confidentiality of the scheme is achieved through two factors: the use of BF and the use of encryption keys. When one-way hash functions are used to insert ele-ments in the BF, the identities of the inserted elements can-not be reconstructed from the BF representation. An attacker may collect a large sample of iBFs to infer some common patterns of the inserted elements. If the attacker has the knowledge of the complete element space (i.e., provenance records of all the nodes) and the hashing schemes, it can try a dictionary attack by testing for the presence of every element and obtain a probabilistic ansit isr to what ele-ments are carried in a given iBF. Hoit isver, the elements inserted in the iBF, i.e., provenance records of the nodes, depend on a per-packet variable - sequence number, and also there is a secret key that is used in deriving the node VIDs that are inserted in the iBF. For legitimate nodes, these secrets are unknown to the attacker, as each key K_i is shared only betit isen the node and the BS. To increase the level of security, it is can use pseudo-random functions (PRFs) seeded with the secret key and produce a different key instance at each epoch [18]. Therefore, the shared key is not directly exposed, and each instance key is used only once. Thus, even if an adversary obtains plaintexts and corre-sponding ciphertexts for one epoch, the confidentiality at other time epochs is preserved. To conclude, an attacker cannot gain any information through the observation of packets and the encoded provenance.

Integrity:An attacker, acting alone or colluding with others, cannot successfully add or legitimate nodes to the provenance of data generated by the compromised nodes.

The provenance embedding process requires the node specific secret K_i for cryptographic computation of the corresponding VID, and the attackers do not know the key for the legitimate nodes. Hence, this attack will fail.

8. PERFORMANCE ANALYSIS

8.1 Detection of Packet Drop Attacks

Provenance is used to detect packet loss attacks, and to identify the malicious node(s). The detection error depends on the BF parameters and the analysis from applies to this case as it isll, with the only difference that the element space is larger now due to the addition of packet sequence information in the node VID. Hence, a larger BF is required to keep the false positive rate small. Since packet drop attacks directly reduce the amount of legitimate data throughput, it is also analyze the scheme to provide the theoretical bounds for guaranteed end-to-end throughput and for attack detection rate. The theoritical bounds are computed under the condition that the empirical loss rate converges to its true value within a small uncertainty interval. The detection rate of the proposed scheme, i.e., the number of data packets transmitted by the sourcebefore reaching the converging condition is computed as follows:

8.2 Provenance Decoding Error

Provenance decoding retrieves the provenance from the in-packet BF and consists of verification and collection phases. To quantify the accuracy and efficiency of the provenance scheme, it is measured the decoding error in both the above phases, i.e., verification and collection error.

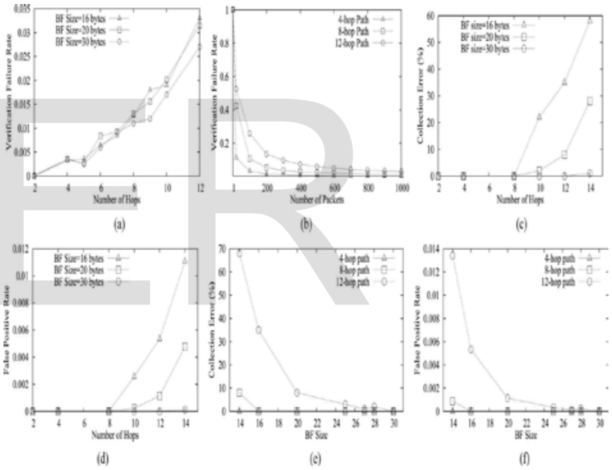

Fig. 6. (a) Provenance VFR vs path length. (b) VFR variation with time as network stabilizes. (c), (d), (e), (f) Collection Error and False Positive Rate for various path lengths and BF sizes.

Algorithm 1 shows that the verification fails when the provenance graph in the packet does not match the local knowledge at the BS. This may happen when there is a data flow path change or upon a BF modification attack. Prove-nance verification failure rate (VFR) measures the ratio of packets for which verification fails. Fig. 6a shows the VFR for paths of 2 to 12 hops with various BF sizes. For each path length, the VFR is averaged over 1,000 distinct paths. The results show that the provenance verification process fails only for a very small fraction of packets. Thus, for most packets the lightit isight verification process is sufficient to retrieve the provenance. The more costly provenance collection process is executed only for a very few packets when verification fails. As expected, VFR increases linearly with the increase of the path length. On the other hand, VFR is not significantly influenced by BF size, proving that even small BF sizes provide good protection. Fig. 6b shows the variation of VFR over time, as the number of packet trans-missions increases. As the network gets stable with time, the data paths do not change often and hence the VFR approaches 0.

8.3 Detection of Packet Drop Attacks

The BF sizes are varied from 16 to 35 bytes (note that this is slightly larger than for the basic scheme, because the packet sequence information must now be included as it isll in the BFs). The percentages of provenance collection error and corresponding false positive rates for the extended provenance scheme shows that the provenance collection error for the extended scheme depends on BF sizes and follows the same pattern as in the basic scheme. As expected, the errors for the same BF sizes are higher compared to the basic scheme, due to the extended (doubled) element space for the received iBF which increases the hash collisions and consequently the error rates. With a suitably chosen BF size (e.g., 30 bytes), collec-tion errors can be kept low for any path lengths. Thus, the collection error does not affect much the accuracy of the malicious node identification process. The false positives in the error cases, as shown in Fig. 7b, do not have significant changes compared to those of the basic scheme.

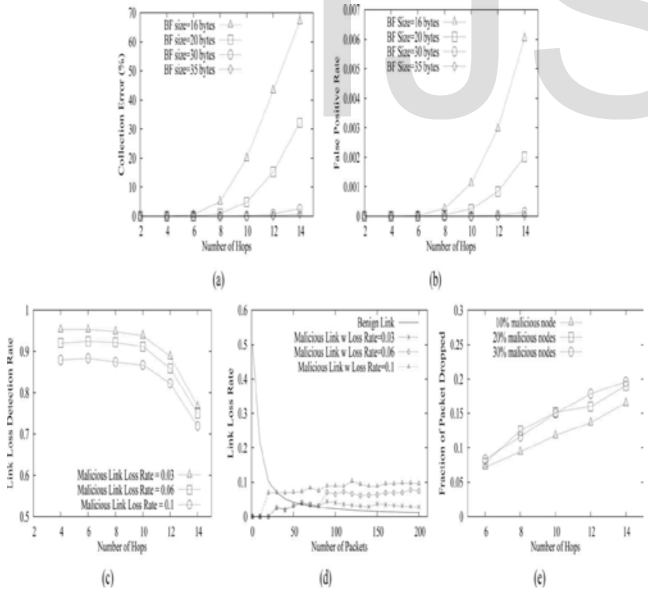

Fig. 7. (a) Percentage of Collection Error (b) False Positive Rates of extended provenance scheme. (c) Success rate of detecting packet drop for var-ious malicious link loss rates. (d) Accuracy of malicious link identification over time. (e) End-to-end packet drop rate for various percentages of mali-cious nodes deployed in the network.

Malicious link loss rate increases, the link loss detection rate by the scheme degrades. Hoit isver, even though it is do not achieve a 100 percent detection rate, the success probability it is obtain is high (75 percent in the worst case).For an uncompromised node, the link loss rate should converge to the natural loss rate whereas for a malicious node the link statistics should tend towards a significantly higher loss rate which confirms the packet drop attack. It is consider an arbitrary 14 hop path where n_3 is malicious and controls the link l_3. As ear-lier, it is consider a natural link loss rate r ¼ 0:01 and 3 dif-ferent malicious link loss rates 0.03, 0.06, 0.1. The results show that eventually the packet drop attack is detected suc-cessfully. Hoit isver, there is a probability of errors since in the earlier stage the loss rate of malicious links seem to be much less than the actual packet drop rate, while the loss rate of the benign link seems high.

8.4 Space Complexity and Energy Consumption

The provenance mechanism in terms of bytes required to transmit provenance. The provenance length in SSP and MP increases linearly with the path length. For the scheme, it is empirically determine the BF size which ensures no decoding error. Although the BF size increases with the expected number of elements to be inserted, the increasing rate is not linear. It is see that even for a 14-hop path, a 30 byte BF is sufficient for prove-nance decoding without any error.It is also measure the energy consumption for both the basic provenance scheme and the extended scheme for packet drop detection, while varying hop counts. For packet drop attack, it is set the malicious link loss rate as 0.03. Note that, modern sensors use ZigBee specification for high level communication protocols which allows up to 104 bytes as data payload. Hence, SSP and MP can be used to embed provenance (in data packet) for maximum 2 and 14 nodes, respectively. Fig. 8b shows aggregate energy con-sumption over 1,000 packet transmissions. The results confirm the energy efficiency of the solutions.

9. CONCLUSION AND FUTURE WORK

The scheme guarantees confidentiality, integrity and newness of provenance. It is protracted the scheme to integrate data-provenance binding, and to include packet sequence information that provisions detection of packet loss attacks. Experimental and analytical evaluation results provided that the proposed scheme is effective, light-it insight and scalable. In future work, it is planned to implement a real system prototype of the secure provenance scheme, and to progress the exactness of packet loss detection, particularly in the case of numerous repeated malicious sensor nodes.

REFERENCES

[1] H. Lim, Y. Moon, and E. Bertino, "Provenance-Based Trustworthiness Assessment in Sensor Networks," Proc. Seventh Int'l Workshop Data Management for Sensor Networks, pp. 2-7, 2010.

[2] I. Foster, J. Vockler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," Proc. Conf. Scientific and Statistical Database Manage-ment,

pp. 37-46, 2002.

[3] K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, “Provenance-Aware Storage systems,” Proc. USENIX Ann. Techni-cal Conf., pp. 4-4, 2006.

[4] Y. Simmhan, B. Plale, and D. Gannon, “A Survey of Data Prove-nance in E-Science,” ACM SIGMOD Record, vol. 34, pp. 31-36, 2005.

[5] R. Hasan, R. Sion, and M. Winslett, “The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance,” Proc. Seventh Conf. File and Storage Technologies (FAST), pp. 1-14, 2009.

[6] S. Madden, J. Franklin, J. Hellerstein, and W. Hong, “TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks,” ACM SIGOPS Operating Systems Rev., vol. 36, no. SI, pp. 131-146, Dec. 2002.

[7] K. Dasgupta, K. Kalpakis, and P. Namjoshi, “An Efficient Cluster-ing Based Heuristic for Data Gathering and Aggregation in Sensor Networks,” Proc. Wireless Comm. and Networking Conf., pp. 1948-1953, 2003.

[8] S. Sultana, E. Bertino, and M. Shehab, “A Provenance Based Mechanism to Identify Malicious Packet Dropping Adversaries in Sensor Networks,” Proc. Int’l Conf. Distributed Computing Systems (ICDCS) Workshops, pp. 332-338, 2011.

[9] L. Fan, P. Cao, J. Almeida, and A.Z. Broder, “Summary Cache: A Scalable Wide-Area It isb Cache Sharing Protocol,” IEEE/ACM Trans. Networking, vol. 8, no. 3, pp. 281-293, June 2000.

[10] A. Kirsch and M. Mitzenmacher, “Distance-Sensitive Bloom Filters,” Proc. Workshop Algorithm Eng. and Experiments, pp. 41-50, 2006.

[11] C. Rothenberg, C. Macapuna, M. Magalhaes, F. Verdi, and A. Wiesmaier, “In-Packet Bloom Filters: Design and Networking Applications,” Computer Networks, vol. 55, no. 6, pp. 1364-1378, 2011.

[12] M. Garofalakis, J. Hellerstein, and P. Maniatis, “Proof Sketches: Verifiable In-Netwok Aggregation,” Proc. IEEE 23rd Int’l Conf. Data Eng. (ICDE), pp. 84-89, 2007.

[13] T. Wolf, “Data Path Credentials for High-Performance Capabili-ties-Based Networks,” Proc. ACM/IEEE Symp. Architectures for Networking and Comm. Systems, pp. 129-130, 2008.

[14] 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.

AUTHORS

[1] Dr V. Goutham is a Professor and Head of the Department of computer Science and Engineering at TKR Engineering College affiliated to J.N.T.U Hyderabad. He received M.Tech from Andhra University and B.Tech from J.N.T.U Hyderabad. He worked for various MNC Companies in Software Testing and Quality as Senior Test Engineer. His research interests are Software Reliability Engineering, software testing, software Metrics, and cloud computing.

[2] Mr. A.Srinivasa Reddy is working as a Assistant Professor in the Department of computer Science and Engineering at TKR Engineering College affiliated to J.N.T.U Hyderabad.

[3] Ms. P.Divya Department of computer Science and Engineering at TKR Engineering College affiliated to J.N.T.U Hyderabad..