IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 01-05
www.iosrjournals.org

Detection of Provenance Forgery And Packet Drop Attacks in Wireless Sensor Networks Using Bloom Filter Algorithm

S.Mailvizhi[1],S.Ramya[2],M..Sreema[3]

[1,2] *UG Students, Department of Computer Science and Engineering, Surya Group of Institutions, Vikiravandi, India*

[3] *Associate Professor, Department of Computer Science and Engineering, Surya Group of Institutions, Vikiravandi, India.*

Abstract: *Large-scale sensor networks are deployed in numerous application domains, and the data they collect are used in decisionmakingfor critical infrastructures .A malicious adversary may introduce additional nodes in the network or compromise existing ones. Therefore, assuring high data trustworthiness is crucial for correct decision-making. Data provenance represents a key factor in evaluating the trustworthiness of sensor data. The proposed technique relies on in- packet Bloom filters to encode provenance. We introduce efficient mechanisms for provenance verification and reconstruction at the base station. In addition, we extend the secure provenance scheme with functionality to detect packet drop attacks staged by malicious data forwarding nodes. We evaluate the proposed technique both analytically and empirically, and the results prove the effectiveness and efficiency of the lightweight secure provenance scheme in detecting packet forgery and loss attacks.*

Keywords: *Provenance, security, sensor networks.*

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 in-network at intermediate hops on their way to a base station (BS) that performs decision-making. The diversity of data sources creates the need to assure the trustworthiness of data, such that only trustworthy information is considered in the decision process. Data provenance is an effective method to assess data trustworthiness, since it summarizes the history of ownership and the actions performed on the data. Recent research 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 modelling, collection, and querying have been studied extensively for workflows and curated databases provenance in sensor networks has not been properly addressed. We investigate the problem of secure and efficient provenance transmission and processing for sensor networks, and we use provenance to detect packet loss attacks staged by malicious sensor nodes. In a multi-hop sensor network, data provenance allows the BS to trace the source and forwarding path of an individual data packet. Provenance must be recorded for each packet, but important challenges arise due to the tight storage, energy and bandwidth constraints of sensor nodes.

Therefore, it is necessary to devise a light-weight provenance solution with low overhead. Furthermore, sensors often operate in an untrusted environment, where they may be subject to attacks. Hence, it is necessary to address security requirements such as confidentiality, integrity and freshness of provenance. Our goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs. 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. We 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. 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 use only fast message authentication code (MAC) schemes and Bloom filters, which are fixed-size data structures that compactly.

II. Literature Review

The method used for data collected in sensor network our approach is based on concept o f provenance as sensor networks are being increasingly deployed in decision-making infrastructures such as battlefield monitoring systems and SCADA (Supervisory Control and Data Acquisition) systems. To obtain trust scores, we propose a cyclic framework which well reflects the inter-dependency property: the trust score of the data affects the trust score of the network nodes that created.

And manipulated. As increasing amounts of valuable information are produced and persist digitally, the ability to determine the origin of data becomes important. In science, medicine, commerce, and government, data provenance tracking is essential for rights protection, regulatory compliance, management of intelligence and medical data, and authentication of information as it flows through workplace tasks, we show how to provide strong integrity and confidentiality assurances for data provenance information.

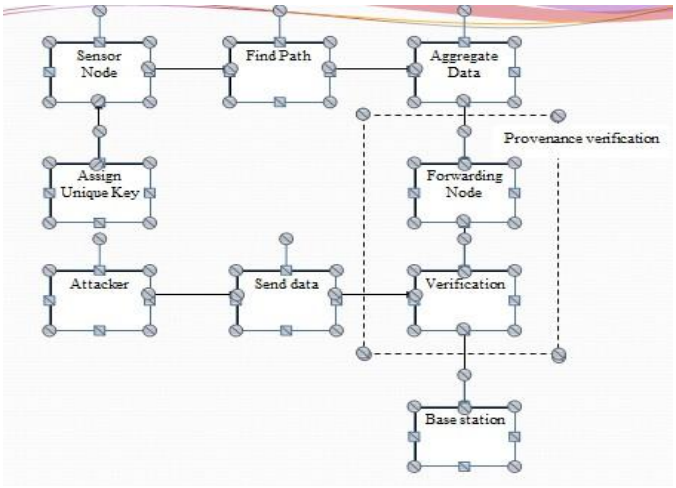
III. Materials And Methods

3.1 Network Model

We considera multichip wireless sensor net work, consisting of a number of sensor nodes and a base station that collects data from the network. The network is modeled as a graph GðN;LÞ, where N ¼ fnij; 1 _ i _ jNjg is the set of nodes, and L is the set of links, containing an element li;j for each pair of nodes ni and nj 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. The BS assigns each node a unique identifier nodeID and a symmetric key Ki. In addition, a set of hash functions H ¼ fh1; h2; . . . ; hkg are broadcast to the nodes for use during provenance embedding.

3.2 Data Model

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



Each data packet contains 1) a unique packet sequence number, 2) a data value, and 3) provenance. We consider 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 GðV;EÞ where each vertex v 2 V is attributed to a specific node HOSTðvÞ ¼ 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.

3.3 Threat Model And Security Objectives:

We assume that the BS is trusted, but any other arbitrary node may be malicious. An adversary can eavesdrop andperform traffic analysis anywhere on the path. In addition, the adversary is able to deploy a few

malicious nodes, as well 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. We 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. Our objective is to achieve the following security properties: 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.

3.4 The Bloom Filter:

A counting bloom filter (CBF) associates a small counter with every bit, which is incremented/decremented upon item insertion/deletion. To answer approximate set membership queries, the distance-sensitive Bloom filter has been proposed. However, aggregation is the only operation needed in our problem setting. The cumulative nature of the basic BF construction inherently supports the aggregation of BFs of a same kind, so we do not require CBFs or other BF variants.

IV. Result And Discussion

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. Similar to the basic scheme in Section 3, the BS first executes the provenance verification process upon receiving a packet. The BS knows 1) the current data path for the packet (decoded from the provenance 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. Thus the verification is bound to fail when pSeq and pSeqb do not match, which also indicates a possible packet loss and suffices to execute provenance collection process directly skipping the verification. Provenance collection. Collection attempts to retrieve the nodes from the encoded provenance, confirm a packet loss and identify the malicious node that dropped the packet. It also distinguishes between the packet drop attack and other attacks that might have altered the iBF. Note that, in case of a path change, the new nodes can be easily learnt through an iteration of ibf membership testing over all the nodes.

4.1security Discussion:

In this section, we discuss the security properties of the proposed provenance scheme. Confidentiality. Claim 1. It is computationally infeasible for an attacker to gain information about the sensor nodes included in the provenance by observing data packets. Justification. 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 elements in the BF, the identities of the inserted elements cannot 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 answer to what elements are carried in a given iBF. However, the elements inserted in the iBF 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 Ki is shared only between the node and the BS. To increase the level of security, we 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 corresponding 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.

4.2 Space Complexity:

To implement SSP, we use SHA-1 (160 bit) for Cryptographic hash operations and the TinyECC library [19] to generate 160-bit digital signatures (ECDSA). The nodeID has length 2 bytes, thus the length of each provenance record is 42 bytes. For MP, we use TinySec library [20] to compute a 4-byte CBC-MAC. Hence, a provenance record has 6 bytes in this case. As each node in the path encodes its own provenance record, the provenance size increases linearly with the number of hops. For a D-hop path, the provenance is 42D

bytes in SSP and 6D bytes in MP.

4.3 Simulation Results:

We implemented and tested the proposed techniques using the TinyOS simulator (TOSSIM) [24]. We have used the micaz energy model and PowerTOSSIM z [25] plug-in to TOSSIM to measure the energy consumption. We consider a network of 100 nodes and vary the network diameter from 2 to 14. All results are averaged over 100 runs. First, we look at how effective the secure provenance encoding scheme (introduced in Section 3) is in detecting provenance forgery and path changes. Next, we investigate the accuracy of the proposed method for detecting packet loss (which was presented in Section 4). Finally, we measure the energy consumption overhead of securing provenance.

4.4 Space Complexity And Energy Consumption:

The provenance length in SSP and MP increases linearly with the path length. For our scheme, we empirically determine the BF size which ensures no decoding error. Although then we 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, we 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. The results confirm the energy efficiency of our solutions.

V. Conclusion

We addressed the problem of securely transmitting provenance for sensor networks, and proposed a light- weight provenance encoding and decoding scheme based on Bloom filters. The scheme ensures confidentiality, integrity and freshness of provenance. We extended the scheme to incorporate data-provenance binding, and to include packet sequence information that supports detection of packet loss attacks. Experimental and analytical evaluation results show that the proposed scheme is effective, light-weight and scalable. In future work, we plan to implement a real system prototype of our secure provenance scheme, and to improve the accuracy of packet loss detection, especially in the case of multiple consecutive malicious sensor nodes.

Acknowledgment

This work was partially funded by US National Science Foundation (NSF) awards CNS-1111512 and CNS - 0964294.

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 Management,
[3]. pp. 37-46, 2002.
[4]. K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, "Provenance-Aware Storage systems," Proc. USENIX Ann. Technical Conf., pp. 4-4, 2006.
[5]. Y. Simmhan, B. Plale, and D. Gannon, "A Survey of Data Provenance in E-Science," ACMSIGMODRecord, vol. 34, pp. 31-36, 2005.
[6]. 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
[7]. Technologies (FAST), pp. 1-14, 2009.
[8]. 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.
[9]. 36, no. SI, pp. 131-146, Dec. 2002.
[10]. K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An Efficient Clustering Based Heuristic for Data Gathering and Aggregation in Sensor Networks," Proc. Wireless Comm. and Networking Conf., pp. 1948-
[11]. 1953, 2003.
[12]. 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.

[13]. L. Fan, P. Cao, J. Almeida, and A.Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol," IEEE/ACM Trans. Networking, vol. 8, no. 3, pp. 281-293, June 2000.
[14]. A. Kirsch andM. Mitzenmacher, "Distance-Sensitive BloomFilters," Proc.Workshop Algorithm Eng. and Experiments, pp. 41-50, 2006.

AUTHORS BIOGRAPHY

S.Ramya is a final year student of Computer Science and Engineering at surya group of institutions, Vikkiravandi. Her area of interest includes Data Mining and Information & Knowledge Management.

S.Mailvizhi is a final year student of Computer Science and Engineering at surya group of institutions, Vikkiravandi. Her area of interest includes Data Mining and Information & Knowledge Management.