Perceive the Packet Drops in Wireless Sensor Networks using New Secure Cuckoo Filter

G. Rajkumar
PG Scholar
Department of Computer Science & Engineering
R.V.S College of Engineering

J. Lourdu Xavier
Assistant Professor
Department of Computer Science & Engineering
R.V.S College of Engineering

Abstract

Bloom Filters are used for high speed set membership tests in large scale sensor networking systems. It allows a small part of false positive answers with very good space effectiveness. However, it does not permit deletion of items from the set, and before attempts to make bigger "standard" Bloom filters to support deletion all degrade of both space or performance. A malicious adversary may initiate further nodes in the network or confrontation previous ones. Therefore, assuring high data responsibility is crucial for correct decision-making. Data provenance represents a key factor in evaluating the responsibility of sensor data. Provenance management for sensor networks introduces more than a few challenging requirements, such as low energy and bandwidth consumption, efficient storage and secure transmission. For the reason propose a novel lightweight scheme to securely transmit provenance for sensor data. And introduce resourceful mechanisms for provenance verification and reconstruction at the base station. In addition, it extends the secure provenance scheme with functionality to detect packet drop attacks staged by malicious data forwarding nodes, propose a new data structure called the cuckoo filter that can replace Bloom filters for approximate set membership tests. Cuckoo filters support adding and removing items dynamically while achieving even higher performance than Bloom filters.

Keywords: Provenance Mechanism, Security Mechanism, Wireless Sensor Networks, Bloom Filter mechanism, Packet forwarding, Cuckoo search algorithm

I. INTRODUCTION

The wireless sensor networks are one of the largest growing technologies in area of data processing and communication networks today. It is a distributed intelligent network system which can accord the environment to complete assigned tasks independently and efficiently due to the vast potential of sensor networks to enable applications that connect the physical world to the virtual world. Each node consists of processing capability using one or more microcontrollers, may contain multiple types of memory, have a RF transceiver, have a power source, and accommodate various sensors and actuators [17]. There are many security schemes have been suggested for WSN, but the choice of security model based on different properties, protection level, and frame formats [10]. The key evaluation metrics for WSN are lifetime, coverage, cost and ease of deployment, response time, temporal accuracy, security, and effective sample rate. The major topics in wireless sensor network security, and present the obstacles and the requirements in the sensor security, classify many of the current attacks, and their corresponding defensive measures. However, the protocol should be resilient against false data injected into the network by malicious nodes and bootstrap secure communication via use of key establishment mechanisms. Different security techniques are introduced for designing cost-efficient, energy efficient protocol for wireless sensor network like. Secure routing, cryptography, access control protocol and dynamic energy based encoding. One of most important research problem in WSN is how to secure the sensor network topology and its communication procedure from potential changes that can be made by malicious nodes inside the existing network. These wireless sensor network are consisting tiny sensor that really suffer the lack of processing capability, memory requirement and battery power utilization. There are many attacks introducing on this infrastructure like Wormhole attack, Sybil attack, selective forwarding, impersonation attack and protocol specific attack

II. BACKGROUND

A. Network Model

We have create a multihop wireless sensor network, consisting of a number of sensor node and a base station that collects data from the network. The networks is modelled as a graph G(N, L), where $N = \{n\ i\ |, 1 \leq i \leq |N\ |\}$ is the set of nodes, and L is the set of link, containing an element l i,j for each pair of nodes n i and n j that are communicating directly with each other. The Base station assigns each node a unique identifier nodeID and a symmetric cryptographic key K i .

B. Data Model

We consider a multiple-round process of collecting data. Each sensor generates data periodically, and individual values are aggregated towards the Base station using any existing hierarchical dissemination scheme. Each data packet contains of (i) a unique packet sequence number, (ii) a data value, and (iii) provenance.

C. Threat Model

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

D. The Bloom Filter (BF)

Several BF variations that provide additional functionality exist. 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.

III. MODULE DESCRIPTION

A. System and Provenance Model

1) System Model

It considers a multihop wireless sensor network, consisting of a number of sensor nodes and a base station that collects data from the network. Sensor nodes are stationary after deployment, but routing paths may change over time. Each node reports its neighboring (i.e., one hop) node information to the BS after deployment. The BS assigns each node a unique identifier node ID and a symmetric cryptographic key Ki.

Most of the current research in wireless sensor networks is constraint driven and focuses on optimizing the use of limited resources (for example, power) at each sensor. While such constraints are important, there is a need for more general performance metrics describing the effectiveness of wireless sensor networks. There is also a need for a unified model that would enable comparison of different types of wireless sensor networks. It proposes a new service-centric model that focuses on services provided by a wireless sensor networks and their corresponding performance metrics.

A wireless sensor networks is modelled at different levels of abstraction. For each level, a set of services and a set of metrics are defined. A mapping between metrics at different levels relates high-level, mission-oriented metrics to low-level capability-oriented metrics. The model consists of mission, network, region, sensor, and capability layers. Within each layer, this planes are identified, namely, communications, management, application and generation learning.

2) Provenance Model

It 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 and for detecting selective forwarding attacks. depicts the extended provenance encoding process, 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 this solution, a node ni creates a vertex vi for every jth packet it generates/forwards. The vertex ID vidi is generated as, where pSeqi is the knowledge of ni about the sequence number of the previous packet in the flow. in updates the provenance of the packet by inserting vidi into the iBF.

Note that, a node must maintain a per-flow record to store the previous packet sequence for each data flow that passed through the node. After a node ni processes/forwards any jth packet, it updates the pSeqi record for the corresponding data flow with the recently processed packet sequence. 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 pSeqi. This addresses the case of routing path changes where a new node in the path can use This special identifier for encoding provenance. Moreover, if a node does not receive packets from a data flow for a long time, it can erase the previous packet information for that flow to reduce space overhead. The node can get updated and maintain this flow-specific record when it receives Packets from that flow more frequently.

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, 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. The provenance verification is performed

according to Algorithm 1, with the only difference that the BS now uses Eq. to create the VID for a node. Verification failure here indicates either a change in the data flow path, a packet drop attack or a BF modification attack, and triggers the provenance collection process.

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. During provenance encoding, every new node in the path uses a special purpose packet identifier (e.g., 0) as the previous packet sequence and generates its VID as. Therefore, to retrieve the new nodes in the path, the decoding scheme at the BS should perform an ibf membership testing over all the nodes, where the VID for each node will be generated using the pre-specified previous packet identifier, along with the node ID and the packet sequence number, seq½j_. For the remainder of the discussion, it assume that a data packet d½j_ has been dropped by an intermediate node ni. Thus, the nodes nl; n1; . . . ; ni received d½j_ and updated their lastly seen packet sequences to seq½j_. On the contrary, nodes niþ1; . . . ; np as well as the BS did not observe d½j_, They have no information to update the preceding packet sequence, and they retain the same old identifier.

B. Cuckoo Filter

The cuckoo filter that can replace Bloom filters for approximate set membership tests. Cuckoo filters support adding and removing items dynamically while achieving even higher performance than Bloom filters. To make cuckoo filters highly space efficient, it use a multi-way associative cuckoo hash table to provide high-speed lookup and high table occupancy (e.g., 95% hash table slots filled); to further reduce the hash table size, each item is first hashed into a constant-sized fingerprint before inserted into this hash table.

Cuckoo filters perform Insert, Lookup and Delete operations at presents partial-key cuckoo hashing, a variant of standard cuckoo hashing that enables cuckoo filters to insert new items dynamically. This technique was first introduced in previous work, but there the context was improving the lookup and insert performance of regular cuckoo hash tables where full keys were stored. In contrast, this paper focuses on optimizing and analyzing the space efficiency when using partial-key cuckoo hashing with only fingerprints, to make cuckoo filters competitive with or even more compact than Bloom filters.

Fig. 1: User video or data page

C. Secure Provenance

Distributed mechanism to encode provenance at the nodes and a centralized algorithm to decode it at the BS. Each packet consists of a unique sequence number, data value, and an iBF which holds the provenance. It emphasize that this focus is on securely transmitting provenance to the BS. This 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.

Provenance is valuable because it allows us to track a result back to its sources. Regardless of how this ancestry is recorded, each relationship reveals information about both parties in the relationship. As it show later, data ancestry can be more or less sensitive than the data itself. Thus provenance security cannot be trivially subsumed by existing security systems.

Provenance has particular characteristics that differentiate it from data typically considered by secure systems. It uses the term traditional data to mean data traditionally considered by secure systems. This includes data stored in file-systems and databases. The sensitivity of provenance and the data it describes may be different.

1) Provenance Encoding

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

Fig. 2: Provenance Encoding

2) Provenance Decoding

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. Afterwards, upon receiving a packet, it is sufficient for the BS to verify its knowledge of provenance with that encoded in the packet.

D. 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.

E. Data-Provenance Binding

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.

If the data aggregation result is verified at the BS, then the data provenance coupling is ensured at each node in the routing path. This objective is to incorporate this provenance scheme with a secure aggregation mechanism so that the aggregation verification process can also be used to check the data-provenance binding.

F. Detecting Packet Drop Attack

It extend the secure provenance encoding scheme to detect packet drop attacks and to identify malicious node (s). It assumes the links on the path exhibit natural packet loss and several adversarial nodes may exist on the path. It augments 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.

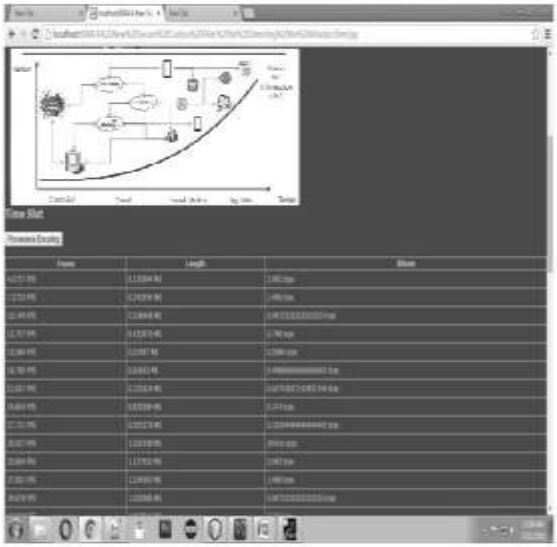

Fig. 3: Packet Loss

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. It utilizes this fact to detect the packet drop attack and to localize the malicious node.

IV. SYSTEM ARCHITECTURE

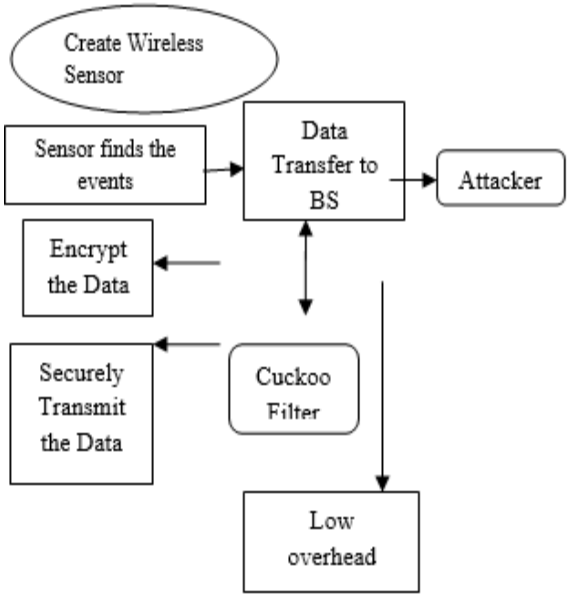

Fig. 4.1: Architecture Diagram

V. CONCLUSION AND FUTURE ENHANCEMENT

Cuckoo filters are a new data structure for approximate set membership queries that can be used for many networking problems formerly solved using Bloom filters. Cuckoo filters improve upon Bloom filters in three ways: (1) support for deleting items dynamically; (2) better lookup performance; and (3) better space efficiency for applications requiring low false positive rates (€ < 3%). A cuckoo filter stores the fingerprints of a set of items based on cuckoo hashing, thus achieving high space occupancy. As a further key contribution, it have applied partial-key cuckoo hashing, which makes cuckoo filters significantly more efficient by allowing relocation based on only the stored fingerprint. This configuration exploration suggests that the cuckoo filter, which uses buckets of size 4, will perform well for a wide range of applications, although appealingly cuckoo filter parameters can be easily varied for application-dependent tuning. The scheme ensures confidentiality, integrity and freshness of provenance. It

extended the scheme to incorporate data-provenance binding, and to include packet sequence information that supports detection of packet loss attacks.

REFERENCES

[1] Amril Syalim, Takashi Nishide, "Preserving Integrity and Confidentiality of a Directed Acyclic Graph Model of Provenance", Proc. Working Conf. Data and Applications Security and Privacy, pp. 311-318, 2010.
[2] Anirudh Ramachandran, Kaushik Bhandankar, "Packets with Provenance", Technical Report GT-CS-08-02, Georgia Tech, 2008.
[3] Chris Karlof and David Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," Proc. Int'l Workshop Sensor Network Protocols and Applications, pp. 113-127, 2003.
[4] Hyo-Sang Lim, Yang-Sae Moon, and Elisa Bertino, "Provenance-Based Trustworthiness Assessment in Sensor Networks," Proc. Seventh Int'l Workshop Data Management for Sensor Networks, pp. 2-7, 2010.
[5] Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Margo Seltzer, "Provenance-Aware Storage systems," Proc. USENIX Ann. Technical Conf., pp. 4-4, 2006.
[6] Nithya N. Vijayakumar, "Towards Low Overhead Provenance Tracking in Near Real-Time Stream Filtering", Proc. Int'l Conf. Provenance and Annotation of Data (IPAW), pp. 46-54, 2006.
[7] Petri Jokela, "LIPSIN: Line Speed Publish/Subscriber" Proc. ACM SIGCOMM Conf. Data Comm., pp. 195-206, 2009.
[8] Ragib Hasan, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance", Proc. Seventh Conf. File and Storage Technologies (FAST), pp. 1-14, 2009.
[9] Salmin Sultana, Elisa Bertino, and Mohamed 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.
[10] Salmin Sultana, Mohamed Shehab, "Secure Provenance Transmission for Streaming Data", IEEE Trans. Knowledge and Data Eng., vol. 25, no. 8, pp. 1890-1903, Aug. 2013.
[11] Stephen Chong, "Self-Identifying Sensor Data", Proc. Ninth ACM/IEEE Int'l Conf. Information Processing in Sensor Networks (IPSN), pp. 82-93, 2010
[12] Tilman Wolf, "Data Path Credentials for High-Performance Capabilities-Based Networks", Proc. ACM/IEEE Symp. Architectures for Networking and Comm. Systems, pp. 129-130, 2008.
[13] Wenchao Zhou, "Secure Network Provenance", Proc. ACMSOSP, pp. 295-310, 2011.