2011 31st International Conference on Distributed Computing Systems Workshops

A Provenance based Mechanism to Identify Malicious Packet Dropping Adversaries in Sensor Networks

Salmin Sultana
Purdue University
ssultana@purdue.edu

Elisa Bertino
Purdue University
bertino@purdue.edu

Mohamed Shehab
University of North Carolina, Charlotte
mshehab@uncc.edu

***Abstract*—Malicious packet dropping attack is a major security threat to the data traffic in the sensor network, since it reduces the legal network throughput and may hinder the propagation of sensitive data. Dealing with this attack is challenging since the unreliable wireless communication feature and resource constraints of the sensor network may cause communication failure and mislead to the incorrect decision about the presence of such attack. In this paper, we propose a data provenance based mechanism to detect the attack and identify the source of attack i.e. the malicious node. For this purpose, we utilize the characteristics of the watermarking based secure provenance transmission mechanism that we proposed earlier and rely on the inter-packet timing characteristics after the provenance embedding. The scheme consists of three phases (i) Packet Loss Detection (ii) Identification of Attack Presence (iii) Localizing the Malicious Node/Link. The packet loss is detected based on the distribution of the inter-packet delays. The presence of the attack is determined by comparing the empricial average packet loss rate with the natural packet loss rate of the data flow path. To isolate the malicious link, we transmit more provenance information along with the sensor data. We present the experimental results to show the high detection accuracy and energy efficiency of the proposed scheme.**

***Index Terms*—Sensor Network, Packet Dropping Attack, Selective Forwarding Attack, Provenance, Inter-packet Delay**

I. Introduction

The proliferation of the internet, embedded systems, and sensor networks has greatly contributed to the wide development of streaming applications. Examples include real-time financial analysis, location based services, transaction logs, sensor networks monitoring environmental characteristics, controlling automated systems, power grids etc. The data that drives such systems is produced by a variety of sources, ranging from other systems down to individual sensors and processed by multiple intermediate agents. This diversity of data sources accelerates the importance of data provenance to ensure secure and predictable operation of the streaming applications. Data **Provenance** is considered as an effective tool for evaluating data trustworthiness, since it summarizes the **history of the ownership and the actions performed** on the data. Recent research works, centered on the provenance-based evaluation of the trustworthiness of sensor data [7], location data [2], and multi-hop network [11], manifest the key contribution of provenance in data streams. Provenance provides the assurance of data trustworthiness, which is highly desired to guarantee accurate decisions in the mission ciritical applications, Supervisory Control And Data Acquisition (SCADA) systems, etc. The significance of provenance for streaming data is also emphasized in the report on *Research and Development Challenges for National Cyber Security* [12] where the research initiatives on efficient and secure implementation of provenance in real-time systems have been highly recommended.

However, existing research on provenance has mainly focused on the tasks of modeling, collection, and querying, leaving the trustworthiness and security issues unexplored. Moreover, although the provenance of workflows, curated databases [5], [8] has been investigated extensively, very few approaches have been reported for data streams. In our work [22], we studied the problem of secure and efficient transmission of provenance in an aggregation supportive streaming environment (focusing on sensor network), which to the best of our knowledge, has not been addressed to such an extent. We propose a framework that transmits provenance along with the sensor data, hiding it over the inter-packet delays (the delay between sensor data items). The embedding of provenance within a host medium makes our technique reminiscent of watermarking [1]. The reason behind adopting watermarking based scheme rather than traditional security solutions like cryptography and digital signature has been discussed elaborately in the piece of work. We also justify the design choices of using *inter-packet delays* (IPD) as the watermark carrier, employing a *spread spectrum* based technique to support multi-user communication over the same medium.

In this paper, we investigate how we can utilize the data provenance to address a critical security attack in the sensor network - the *malicious packet dropping attack*. In this attack, a malicious node drops a packet to prevent its further propagation. To avoid raising suspicions, the adversary does not drop every packet, instead selectively drops packets and forwards the remaining traffic. For this nature, the attack is also known as *Selective Forwarding Attack*. The mass deployment of the tiny sensors, often in unattended and hostile enviroments makes them susceptible to such data plane attacks. The dropping attack can result a significant loss of (sensitive) data and degrade the legitimate network throughput.

1545-0678/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCSW.2011.54

Thus, packet dropping attack constitutes a major security threat to the data traffic in the sensor network. However, the attack is hard to detect, since the lack of reliability in wireless communication and transient network congestion can cause packet losses. Furthermore, typical sensor nodes are resource (energy, bandwitdth etc.) constrained. The power scarcity can make a node unavailable, whereas there may be communication failure due to some reasons, such as, physical damage etc. These factors can also result packet drop. Hence, it may be difficult to identify whether a packet drop is caused by the selective forwarding attack or non-security reasons.

A common approach to defend the packet drop attack is *multipath routing* [16], [17]. But, multipath routing suffers from several drawbacks such as, high communication overhead with the increase of the number of paths, inability to identify the malicious node etc. Traditional transport layer protocols [13], [14] for sensor networks also fail to guarantee that packets are not maliciously dropped. They are not designed to deal with the malicious attacks.

In this paper, we propose a provenance based mechanism to handle the packet dropping attack. The key point is that, we utilize the *inter-packet delay* based provenance transmission technique that we proposed [22] and devise a detection mechanism based on the distribution of these delays. Thus, the detection technique is integrated with our provenance transmission mechanism and now one solution serves the dual purpose. To isolate the malicious node, we transmit more provenance information and use the data channel for this purpose. These characteristics ensure the light weight of our scheme. The contributions of this paper include *(i) addressing the problem of malicious packet dropping attack in sensor network, (ii) design of a provenance based approach for detecting the attack and then isolating the malicious node (v) an experimental evaluation exhibiting the high accuracy and energy efficiency of the scheme.*

The rest of the paper is organized as follows: Section II explains the system model and the preliminaries. Provenance encoding and decoding process according to our scheme are summarized in Section IV. Section V describes the proposed scheme for packet drop attack detection. Experimental results are presented in Section VI. Section VII summarizes the related work. Finally, we conclude in Section VIII.

II. System Model

A. Network and Communication Model

Network Model: We consider a typical deployment of wireless sensor networks, consisting of a large number of nodes. Sensor nodes are stationary after deployment, though routing paths may change due to node failure, resource optimization, etc. The routing infrastructure is assumed to have a certain lower bound on the time before the routing paths change in the network. The network is modeled as a graph G (*N,E*) where
- *N* = $\{n_i : n_i$ is a network node with identifier i$\}$: a set of network nodes
- *E* = $\{e_{i,j} : e_{i,j}$ is an edge connecting nodes n_i and $n_j\}$: the set of edges between the nodes in *N*

There exists a powerful base station (BS) that acts as sink/root and connects the network to the outside infrastructure such as Internet. All nodes form a tree rooted at the BS and report the tree topology to BS once after the deployment or any change in topology. Since the network does not change so frequently, such a communication will not incur significant overhead.

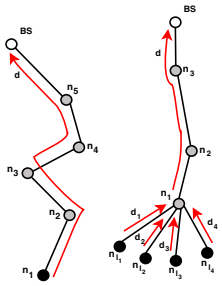
The network is organized into a cluster structure according to some well known dynamic clustering algorithms, such as LEACH [6]. Sensory data from the children are aggregated at cluster-head a.k.a. Aggregator and routed to the applications through the routing tree rooted at the BS. Here, we do not assume anything regarding the clock synchronization between nodes.

Communication Model: We assume that BS cannot be compromised and it has a secure mechanism (e.g., μTESLA [9]) to broadcast authentic messages into the network. Each sensor has a unique identifier and shares a secret key K_i with the BS. Each node is also assigned a *Pseudo Noise (PN) sequence* of fixed length L_p which acts as the provenance information for that node.

B. Data Model

The sensor network allows multiple distinguishable data flows where source nodes generate data **periodically**. A node may also receive data from other nodes to forward towards the BS. For the rest of the paper, we will use *data arrival* to imply data generation or receipt at a node. While transmitting, a node may send the sensed data, may pass an aggregated data item computed from multiple sensors' readings or act as an intermediate routing node. Each data item (packet) in the transmission stream contains an attribute value and provenance for that attribute. The data packet is also timestamped by the source with the generation time. As we see later, packet timestamp is crucial for provenance embedding and decoding process and hence we use Message Authentication Code (MAC) to maintain its integrity and authenticity. The MAC is computed on data value and timestamp to ensure the same properties for data.

Provenance: The provenance of a data item includes information about its origin and how it has been transmitted to the BS. The notion of provenance is formally defined as follows.
Definition. *The provenance p_d for a data item d is a rooted tree satisfying the properties: (1) p_d is a subgraph of the sensor network G(N,E); (2) the root node of p_d is the BS, expressed as n_b; (3) for $n_i, n_j \in$ N included in p_d, n_i is a child of n_j if and only if n_i participated in the distributed calculation of d and/or passed data information to n_j.*
Figure 1 shows two different kinds of provenance. In Figure 1(a), data item *d* is generated at leaf node n_1 and the internal nodes simply pass it to BS. We call such internal nodes *Simple Node* and this kind of provenance *Simple Provenance*. The simple provenance can be represented as a simple path. In Figure 1(b), the internal node n_1 generates the data *d* by aggregating data $d_1, ..., d_4$ from $n_{l_1}, ..., n_{l_4}$ and passes *d* towards BS. Here, n_1 is an aggregator and the provenance is called *Aggregate Provenance*, which is represented as a tree.



(a) Simple Provenance (b) Aggregate Provenance

Fig. 1. Provenance examples for a sensor network

C. Adversary Model

We assume that the source and the destination node (i.e. the BS) on the path being monitored are honest. An adversary is in complete control of an arbitrary number of intermediate nodes on the path, including the knowledge of their secret keys. The adversary can eavesdrop and perform traffic analysis anywhere on the path. The adversary may drop, inject or alter packets on the links that are under its control. Still the adversary cannot influence the natural packet loss rate on the links on the path.

Our goal is twofold -

- To transfer the provenance along with sensor data in a bandwidth efficient manner while ensuring data quality. We propose a distributed strategy to securely embed provenance as a list of unordered nodes over the inter-packet delays. Upon extracting the list of nodes, the BS can easily determine their order with the knowledge of network topology and construct the provenance tree. It is important to mention that the information to include in provenance actually depends on the application. The usage of provenance for computing trust scores of data items according to the scheme proposed by Lim et al. [7] shows, for example, that for assessing the sensor data trustworthiness it is often sufficient to know the set of nodes associated with a data flow. Hence, our provenance embedding and decoding scheme can also be utilized without any additional knowledge.
- To detect the packet dropping attack and then identify the adversary utilizing this provenance transmission technique.

In this paper, we are focused to achieve the second objective.

III. Background

Utilizing the *spread spectrum watermarking* to pass on the provenance of multiple sensor nodes over the same medium is a key design choice in our proposed scheme for secure provenance transmission. Hence, we provide a brief description of *spread spectrum watermarking* to facilitate the understanding of the scheme.

Spread Spectrum Watermarking: Spread spectrum is a transmission technique where a narrowband data signal is *spread* over a much larger bandwidth so that the signal energy present in any single frequency is undetectable [3]. In our context, the *set of inter-packet delays (IPDs)* is considered as *Communication Channel* and *provenance* is the *Signal* transmitted through it. Provenance is spread over many IPDs such that the information present in one container is small. Consequently, any unauthorized party needs to add high amplitude noise to all of the containers to destroy provenance. Thus, the use of the spread spectrum technique for watermarking provides strong security against different attacks. We have adopted *Direct Sequence Spread Spectrum* (DSSS) in our scheme since it is widely used for enabling multiple users to transmit simultaneously on the same frequency range by utilizing distinct pseudo-noise(PN) sequences [3]. The intended receiver can extract the desired user's signal by regarding other signals as noise-like interferences. The components of DSSS system are

Input:

- The original data signal *d(t)*, as a series of +1 and -1.
- A PN sequence *px(t)*, encoded like the data signal. N_c is the number of bits per symbol and is called PN length.

Spreading: The transmitter multiplies data with PN code to produce spreaded signal as *s(t) = d(t) px(t)*

Despreading: The received signal *r(t)* is a combination of the transmitted signal and noise in the communication channel. Thus *r(t) = s(t) + n(t)*, where *n(t)* is a white Gaussian noise. To retrieve the original signal, the correlation between *r(t)* and the PN sequence $pr(t)$ at the receiver is computed as $R(\tau) = \frac{1}{N_c}\sum_{t=T}^{T+N_c} r(t)\,pr(t+\tau)$. Now, if px(t) = pr(t) and also $\tau = 0$ i.e. px(t) is synchronized with pr(t), then the original signal can be retrieved. Otherwise, the data signal cannot be recovered. So, a receiver without having PN sequence of the transmitter cannot reproduce the originally transmitted data. This fact is the basis for allowing multiple transmitters to share a channel. In this paper, we'll refer to $R(0)$ by the term *cross-correlation*.

In case of multiuser communication in DSSS, spreaded signals produced by multiple users are summed up and transmitted over the channel. To retrieve the signal for j-th user, the cross-correlation between $r(t)$ and $px_j(t)$ is computed. Multiuser communication introduces noise to the signal of interest and interfere with the desired signal in a proportion to the number of users.

IV. Overview of Secure Provenance Transmission Mechanism

In this section, we provide a overview of our proposed solution for securely transmitting the provenance of sensor data. For the detailed description of the scheme, we refer the readers to our previous work [22].

We propose a novel approach to watermark provenance over the delay between consecutive sensor data items. A set of $(L_p + 1)$ data packets is used to embed provenance over the inter-packet delays. Thus, the sequence of L_p IPDs, $\mathcal{DS} = \{\, \Delta[1], \Delta[2], ..., \Delta[L_p] \,\}$ is the medium where we hide provenance. $\Delta[j]$ represents the inter-packet delay between *j-th* and *(j+1)-th* data item. The process also uses the secret

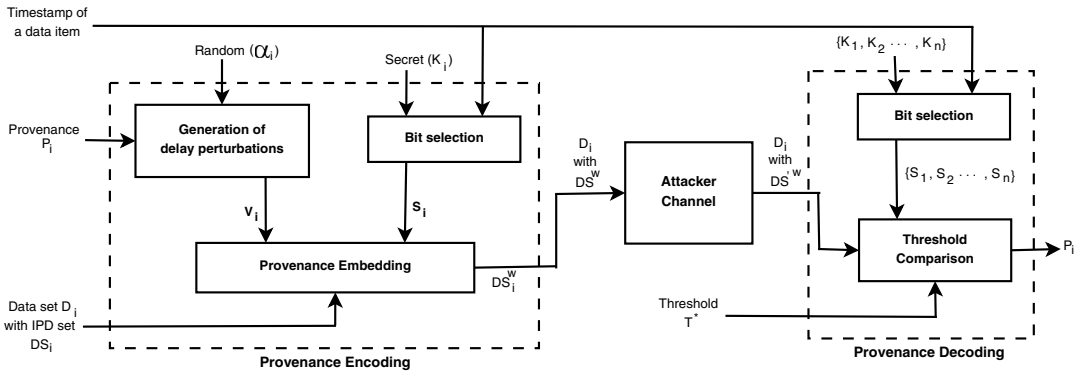


Fig. 2. Stages of Provenance Encoding at the Sensor Node and Decoding at the Base Station.

key K_i ($1 \leq i \leq n$, where n is the number of nodes in the network), a locally generated random number α_i (known as impact factor) and the provenance information $\mathbf{pn_i}$. α_i is a random (real) number generated according to a normal distribution $N(\mu, \sigma)$. μ and σ are pre-determined and known to the BS and all the nodes. The PN sequence consists of a sequence of +1 and -1's and is characterized by a zero mean. The provenance encoding process at a node n_i is summarized below

Step E1. Generation of Delay Perturbations: By using provenance information $\mathbf{pn_i}$ and impact factor α_i, the node generates a set of delay perturbations, $\mathcal{V}_i = \{v_i[1], v_i[2], ..., v_i[L_p]\}$, as a sequence of real numbers. Thus

$$\begin{aligned}\mathcal{V}_i &= \{v_i[1],\, v_i[2],\, ...,\, v_i[L_p]\} \\ &= \alpha_i \times \mathbf{pn_i} \\ &= \{\ (\alpha_i \times pn_i[1]),\, ...,\, (\alpha_i \times pn_i[L_p])\ \}\end{aligned}$$

Note that, $v_i[j]$ corresponds to the provenance bit $pn_i[j]$.

Step E2. Bit Selection: On the arrival of any (j+1)-th data packet, the node records the IPD $\Delta[j]$ and assigns a delay perturbation $v_i[k_j] \in \mathcal{V}$ to it. The selection process uses the secret K_i and packet timestamp $ts[j+1]$ as follows

$$selection(\Delta[j]) = H(K_i \,\|\, (H(ts[j+1] \,\|\, K_i))\ mod\ L_p$$

Here, H is a lightweight, secure hash function, $\|$ is the concatenation operator.

Step E3. Provenance Embedding: In this step, IPD $\Delta[j]$ is increased by $v_i[k_j]$ time unit. As $v_i[k_j]$ corresponds to provenance bit $pn_i[k_j]$, through this step a provenance bit is embedded over an IPD. This notion makes our scheme reminiscent of watermarking. Provenance bits are watermarked over IPDs by manipulating them with corresponding delay perturbations, termed as *watermark delay*. This way, $\mathcal{DS}$ is transformed into the watermarked version $\mathcal{DS}^w$.

The sensor dataset is transmitted towards the BS while reflecting the watermarked IPDs. Throughout the propagation, each intermediate node watermarks its provenance as increasing the IPDs. Data packets may also experience different propagation delays or attacks aimed at destroying the provenance information. At the end, the BS receives the dataset along with watermarked IPDs $\mathcal{DS}^w$, which can be interpreted as the sum of delays imposed by the intermediate nodes, attackers and difference between consecutive propagation delays along the data path. Thus, $\mathcal{DS}^w$ represents the DSSS encoded signal in our context. The provenance retrieval process at the BS approximates the provenance from this DSSS signal based on an optimal threshold T^*. The threshold, corresponding to the network diameter and PN length, is calculated once after the deployment of the network. For retrieval purposes, the BS also requires the set of secret keys $\{K_1, K_2, ..., K_n\}$ and PN sequences $\{\mathbf{pn_1}, \mathbf{pn_2}, ..., \mathbf{pn_n}\}$. The retrieval process at the BS follows two steps:

Step R1. Bit Selection: The IPDs for the incoming packets are recorded at the BS. For each node, the IPDs are reordered according to the algorithm used in **E2**, which produces a node specific sequence $\mathbf{CS_i}$.

Step R2. Threshold based Decoding: For any node n_i, the BS computes the cross-correlation R_i between $\mathbf{CS_i}$ and provenance $\mathbf{pn_i}$ and takes decision on whether $\mathbf{pn_i}$ was embedded by a comparison of R_i with threshold T^*.

As the BS does not know which nodes participated in the data flow, it performs the *Bit selection* and *Threshold Comparison* for all nodes. Based on the threshold comparison result, it deduces the participation of nodes in a data flow.

V. Packet Dropping Adversary Identification Scheme

Utilizing the provenance transmission technique, we propose a method to detect malicious packet dropping attack and identify the malicious node/link. The scheme relies upon the distribution of provenance embedded inter-packet delays and consists of the following phases

- Packet Loss Detection

- Identification of Attack Presence
- Localizing the Malicious Node/Link

The BS initiates the process for each sensor data flow by leading the packet loss identification mechanism based on the inter-packet delays of the flow and the extracted provenance. A packet loss may be due to transient congestion or malicious packet dropping, hence, the BS waits until a sufficient number of packet losses and then calculates the average packet loss rate. A comparison of this loss rate with the natural packet loss rate of the path confirms the event of malicious packet dropping. If a packet drop attack is suspected, the BS signals the data source and the intermediate nodes to start the mechanism for pinpointing the malicious node/link. The details of the scheme is presented below

A. Packet Loss Detection

We use the provenance as well as the carrier IPDs to detect a packet loss. The BS can observe the data flows transmitted by all of the sensor nodes and obtain the timing characteristics for them. Since provenance is embedded over the IPDs, the watermarked IPDs follow a different distribution than the regular (unwatermarked) IPDs. After the receipt of a few initial group of $(L_p + 1)$ packets from a data flow, the BS can approximate the distribution (mean and variance also) of the watermarked IPDs for that flow. Afterwards, the BS analyzes each IPD to check whether it follows the estimated distribution. In the case of data dropping by a node that must be traversed to reach the BS, the attack will end the journey of the data. Here, the IPD observed by the BS will be large enough to go beyond the distribution and be detected as a packet loss. For the packets containing sequence numbers, any out of order packet can verify the detection of packet loss. On the other hand, if the data packet is dropped by an intermediate router within a cluster, it cannot interfere with the data from other nodes to be aggregated at the cluster head and transmitted towards the BS. Consequently, the IPD based check will not be effective in such attack scenerios. Fortunately, in this case, the provenance retrieved at the BS does not include the simple path containing the malicious node (from the source upto the aggregator). Thus, the dissimilarity of this provenance with the provenance of earlier rounds exposes the fact of a packet loss.

B. Identification of Attack Presence

Since the packet loss may be due to transient network congestion or malicious dropping attack, the BS needs to observe the network and transmission chracteristics more to be certain about the malicious attempts. For this purpose, the BS collects G_a group of $(L_p + 1)$ packets, where $G_a > 0$ is a real number. Assume, the BS identifies m packet losses. Thus, the average packet loss rate L_{avg} can be calculated as

$$L_{avg} = \frac{m}{G_a * (L_p + 1) + m}$$

The natural packet loss rate is calculated by

$$L_n = \sum_{i=1}^{h} \rho_i \prod_{j=1}^{i-1} (1 - \rho_j)$$

where, h is the number of hops in the path and ρ_i is the natural loss rate of the link between nodes n_i and n_{i+1}. By having a comparison of L_{avg} with L_n, the BS comes to a decision about the packet dropping attack. If $L_{avg} > L_n$, the BS gets confirmed about the malicious node(s) in the flow path that selectively drops packets.

C. Localizing the Malicious Node/Link

For the purpose of identifying the malicious link, we include more information in the provenance besides the nodeID. The data payload (rather than the timing channel) is used to carry this additional provenance data. Whenever the BS detects the attack, it notifies the source and intermediate nodes in the path about the fact. Afterwards, while sending/forwarding a data packet, each node adds information including the hash of the data, timestamp etc. of the last data packet it recieved through this path. Hence the format of a data packet at a node n_i is

$$m^t = < data \,||\, timestamp \,||\, \mathcal{P}_i >$$
$$\mathcal{P}_i = \{n_i \,||\, H(m^{t-1}) \,||\, \mathcal{P}_{i-1}\}_{K_i}$$

Here, m^t represents the current data packet and m^{t-1} is the most recent packet before the current one. $\mathcal{P}_i$ denotes the provenance report at the node n_i, H is a collision resistant hash function and $\{D\}_K$ denotes a message D authenticated by a secret key K_i using a message authentication code (MAC). As said earlier, the data and timestamp are authenticated using MAC to ensure integrity.

Upon receiving a data packet containing the hash chain of provenance, the BS can sequentially verify each provenance report embedded in it. Assume, the flow path i.e. the data provenance is $\{n_s, n_2, ..., n_{i-1}, n_i, ..., BS\}$, where $n_1 = n_s$ is the source node and $n_d = BS$ is the base station. The link between the nodes n_i and n_{i+1} is represented as l_i. For some $i < d$, if the provenance report from each intermediate node $n_j, j \in [1, i]$ contains the recent value for timestamp and hash of the data but the provenance reports from n_{i+1} contains the older values, then the BS identifies the link l_i as faulty. After observing the G_l group of $(L_p + 1)$ packets, the BS calculates the average packet loss rate of all the links of the path. G_l is also a real number greater than 0. If the loss rate of a link l_i is significantly higher than the natural packet loss rate ρ_i, then the BS convicts the link as a malicious link. To quantify the term *significantly*, we introduce a per-link drop rate threshold, denoted by τ, where $\tau > \rho_i$. If the empirical packet loss rate of a link l_i is greater than τ, then l_i is identified as a malicious link.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme through simulation. The reported results reveal the

accuracy and energy efficiency of the scheme in case of detecting packet dropping attack in the sensor network.

A. Adversarial Setting

In a packet dropping attack, an adversary usually directly compromises a node and selectively drops the packets flowing through that node. We emulate such a realistic scenario by setting malicious nodes in the path to perform malicious packet dropping activity. Here, we deploy exactly one malicious node on the path since such a setting achieves the adversarys optimal strategy [20]. Note that, if a malicious node drops packets while data forwarding towards the BS, it can manifest high drop rate only on its upward adjacent link. Without loss of generality, we assume that, if the malicious node receives but drops a data packet, while forwarding the next data packet it will still embed the additional provenance report honestly as if it were functioning correctly. In this way, a malicious node n_is dropping activity always increases the drop counts of its upward adjacent link l_i. Therefore l_i is the target to identify.

B. Network Topology and Parameters

For experiments, we simulate the sensor network as a tree with diameter (number of hops in a path) H and F children per node. The network consists of 1000 nodes with default values of H = 8, F = 8 and L_p = 160 bits. Sensor data is 16 bits and generated at every 5 sec. According to the the CC2420 stack implementation in tinyOS, transmitting a data packet (in our case) takes 2.56 ms with 250 Kbit/s transmission rate. Considering congestion back-off time of 0 to 2.5 ms, ACK duration 3.84 ms and 2 μs propagation delay, the total time for a packet delivery takes 6.4 to 8.9 ms. Thus, one hop packet delivery times vary within a [0, 2.5] ms range. We consider this variation in transmission time in our simulation.

The simulation model considers a single data flow and one flow path with a stationary data source and the BS as the destination. We simulate the scheme on one path with varying locations of the malicious link, but find that the results do not depend on the location of the adversary. Hence, we fix the malicious node and so as the malicious link. All of the links are assigned the benign per-link loss rate $\rho = 0.01$ [20]. We set the malicious per-link drop rate = 0.03 i.e. the drop rate of the malicious node is 0.02 and the per-link drop threshold, τ = 0.02. Each packet traversing a link (or the malicious node) has an independent probability of being dropped below the corresponding drop rate threshold of that link (or the malicious node).

C. Simulation Results and Analysis

We evaluate (a) the attack and adversary detection rate and (b) average energy consumption at a node for the proposed scheme. We run the simulation 100 times for each experiment to calculate the averaged results.

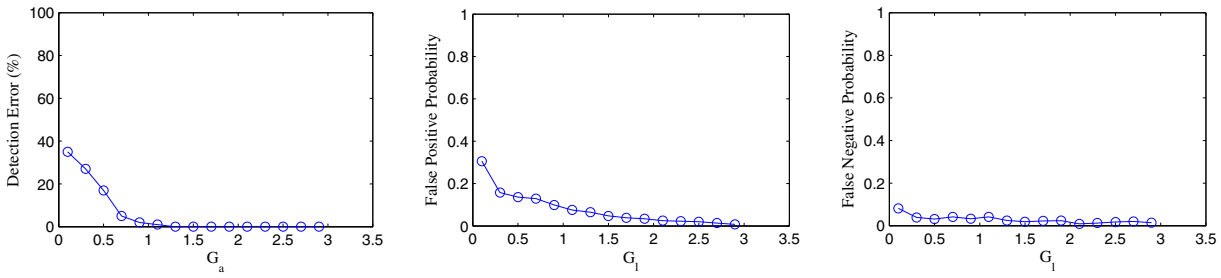
1) High Detection Rate: We show the accuray of our scheme from two perspectives (i) effectiveness of the scheme in detecting the presence of the packet drop attack (ii) identifying the malicious link. For this purpose, we measure the detection error of our scheme for various number of packets observed by the BS. Since, the packet loss may increase due to transient network congestion or other kind of communication failures, observing a small number of packets may often lead to a wrong decision about the attack presence. Hence, we vary the number of packets that the BS considers by adjusting the parameters G_a (in case of attack detection) and G_l (in case of identifying the attack source). The values of G_a and G_l are proportional to the number of packets observed since the BS utilizes the statistics of a G_a (or G_m) group of packets for the detection mechanisms. Figure 3(a) reports the detection error of our scheme in case of identifying the packet drop attack. Predictibly, an increase in G_a decreases the detection error. In case of localizing the malicious node, we report the false positive probability 3(b) and false negative probability 3(c). We also notice the same trend in these results with the increase of G_l. Since, the network congestion or other source of communication failures may increase the natural loss rate of a link significantly, we see a high false positive probability for smaller values of G_l.

2) Energy Efficiency: Since energy is a major constraint in the sensor network, we evaluate the energy cost at a node due to running the protocol. In the first two phases, the sensor nodes do not have to perform any additional task. In case of identifying the malicious link, each node in the path has to pass on more provenance information and transmit an authenticated provenance report along with the data. Hence, we recorded the energy comsumtption at a node in the malicious link identification phase. Our experimental results show that the power usage increases only by 0.06% in our case. The power usage is almost similar for any network diameter and any number of packets observed. Hence, the proposed mechanism is light weight.

VII. Related Work

There has been a lot of research efforts to explore various mechanisms for handling the malicious data drop attack. These mechanisms can be classified into the following categories multipath routing protocols, acknowledgement based mechanisms, protocols using specialized hardware.

The multiplath routing protocols [16], [17] first discover multiple paths for data forwarding and then uses these paths to provide redundancy in the data transmission from a source. The data is encoded and divided into multiple shares and then sent to the BS via different routes. However, these methods can not identify the malicious node. They increase the network flow significantly, hence are not suitable for the resource constrained sensor networks. Additionally, these mechanisms could be vulnerable to route discovery attacks that prevent the discovery of non-adversarial paths. Examples of protection mechanisms that require specialized hardware include [18], and [19]. The authors in [18] introduce a scheme called packet leashes that uses either tight time synchronization or location awareness through GPS hardware. The work in [19] relies on hardware threshold signature implementations to prevent one node from propagating errors or attacks in the whole network.



(a) (b) (c)

Fig. 3. (a) Percentage error in detecting the presence of attack. The number of packets observed by the BS = $G_a * (L_p + 1)$. (b) Rate of False positives, and (c) False negatives in case of detecting the malicious node. G_l is also proportional to the number of packets the BS considers

The acknowledgement based protocols [20], [21] expect the authenticated acknowledgement from the intermediate nodes and the BS within a certain time. This method would render malicious packet dropping detectable at the end points (data source or the BS). However, the method incurs high communication overhead and in some cases has to be augmented with other techniques for diagnosis and isolation of the malicious nodes.

VIII. Conclusion

In this paper, we address the problem of malicious packet dropping attack in sensor network and propose a provenance based mechanism utlizing our provenance transmission method. Experimental results show that the scheme is able to detect the attack with high accuray and energy efficiency. In future, we intend to utilize the provenance to address and mitigate other security attacks for sensor network. We also plan to develop a comprehensive taxonomy of provenance data and investigate the validation of the proposed technique for large sized provenance.

IX. Acknowledgement

The work reported in this paper has been partially by Northrop Grumman as part of the NGIT Cybersecurity Research Consortium and by NSF under grant CNS-0964294.

References

[1] I. J. Cox and M. L. Miller. The first 50 years of electronic watermarking. *J. Appl. Signal Process.*, vol. 2, pp. 126-132, 2002.

[2] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu. An Approach to Evaluate Data Trustworthiness Based on Data Provenance. *Intl. Workshop on Secure Data Management*, pp. 82-98, 2008.

[3] R. Dixon. Spread Spectrum Systems. John Wiley & Sons, 1984.

[4] R. Hasan, R. Sion, and M. Winslett. The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance, *USENIX Conf. on File and Storage Technologies* (FAST), pp. 1-14, 2009.

[5] I. Foster, J. Vockler, M. Wilde, Y. Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. *Intl. Conf. on Scientific and Statistical Database Management*, pp. 37-46, 2002.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Effcient Protocol for Wireless Micro sensor Networks. *Hawaii Intl. Conf. on System Sciences*, pp. 3005-3014, 2000.

[7] H. Lim, Y. Moon, and E. Bertino. Provenance-based trustworthiness assessment in sensor networks. *Workshop on Data Management for Sensor Networks*, pp. 2-7, 2010.

[8] K.-K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer. Provenance aware storage systems. *USENIX Annual Technical Conf.*, 2006.

[9] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: security protocols for sensor netowrks. *Wireless Networks*, pp. 189-199, 2001.

[10] Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. *SIGMOD Record*, Vol. 34, pp. 31-36, 2005.

[11] X. Wang, and P. Mohapatra. Provenance Based Information Trustworthiness Evaluation in Multi-hop Networks, *IEEE GLOBECOM*, 2010.

[12] National Cyber Security Research and Development Challenges, 2009 Related to Economics, Physical Infrastructure and Human Behavior, 2009

[13] C. Y. Wan, A. T. Campbell, L. Krishnamurthy. PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks. *The first ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1-11, 2002.

[14] Y. Sankarasubramaniam, O. B. Akan and I. F. Akyildiz. ESRT: Event to Sink Reliable Transport in Wireless Sensor Networks. *ACM MobiHoc*, pp. 177-188, 2003.

[15] B. Deb, S. Bhatnagar, and B. Nath. ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks. *IEEE Local Computer Networks*, 2003, pp: 406-415.

[16] S.J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. *IEEE International Conference on Communications (ICC)*, pp. 3201-3205, 2001.

[17] Y. Lu, V.Wong. An energy efficient multipath routing protocol for wireless sensor networks. *International Journal of Communication Systems*. 2007, pp:747-766

[18] Y. C. Hu, A. Perrig, and D.B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. *IEEE INFOCOM*, 2003, pp. 1976-986

[19] C. Basile, Z. Kalbarczyk, and R. K. Iyer. Neutralization of Errors and Attacks in Wireless Ad Hoc Networks. *DSN* 2005, pp. 518-527.

[20] X. Zhang, A. Jain, and A. Perrig. Packet-dropping Adversary Identification for Data Plane Security. *The 4th ACM SIGCOMM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Madrid, Spain, December, 2008.

[21] B. Carbunar, I. Ioannidis and C. Nita-Rotaru. JANUS: Towards Robust and Malicious Resilient Routing in Hybrid Wireless Networks. *WiSe* 2004, pp. 11-20.

[22] S. Sultana, M. Shehab, E. Bertino. Secure Provenance Transmission for Streaming Data. SUBMITTED in *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 2011.