Perfect Contextual Information Privacy in WSNs under Colluding Eavesdroppers

Alejandro Proaño
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, USA
aaproano@ece.arizona.edu

Loukas Lazos
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, USA
llazos@ece.arizona.edu

ABSTRACT

We address the problem of preserving contextual information privacy in wireless sensor networks (WSNs). We consider an adversarial network of colluding eavesdroppers that are placed at unknown locations. Eavesdroppers use communication attributes of interest such as packet sizes, inter-packet timings, and unencrypted headers to infer contextual information, including the time and location of events reported by sensors, the sink's position, and the event type. We propose a traffic normalization technique that employs a minimum backbone set of sensors to decorrelate the observable traffic patterns from the real ones. Compared to previous works, our method significantly reduces the communication overhead for normalizing traffic patterns.

Categories and Subject Descriptors

C.2.0 [**Computer - Communication Networks**]: General - Security and Protection

Keywords

Eavesdropping, colluding adversaries, wireless sensor networks, algorithms, security

1. INTRODUCTION

Wireless communications are vulnerable to eavesdropping by anyone equipped with a wireless receiver. When the transmitted information is of sensitive nature, its privacy is protected via cryptographic methods. However, encryption alone cannot prevent the leakage of contextual information such as the location of communicating nodes, the path between the source and the destination, or the time of occurrence of a reported event. Passive eavesdroppers can obtain contextual information by performing traffic analysis using low-level packet identifiers such as packet size and inter-packet timings, even when the contents of the packet remain hidden [4, 6, 9]. Moreover, this information can be used to launch intelligent attacks of selective and adaptive nature that degrade network performance at low cost [10,13].

In this paper, *we address the problem of preserving the privacy of contextual information in wireless communications.* Though we study this problem in the context of wireless sensor networks (WSNs), our methods are applicable to any static wireless multihop network. We consider an adversary that deploys a network of colluding eavesdroppers at unknown locations within the WSN. The eavesdropping devices can be cheap passive sensors that form an out-of-band collusion network [9, 14]. Eavesdroppers extract communication attributes of interest and centrally process them to derive contextual information.

State-of-the art techniques for hiding contextual information employ bogus transmissions to normalize the eavesdropped transmission patterns [9, 12, 14]. In these schemes, sensors transmit according to a predefined distribution, irrespective of their real traffic profile. Transmissions of real packets conform to the same distribution, thus defeating traffic analysis techniques. However, when the locations of the colluding eavesdroppers are unknown, privacy can be achieved only if all sensors become sources of bogus traffic [9, 12]. In our approach, we significantly reduce the communication overhead by intelligently selecting the bogus sources and loosely coordinating real packet transmissions.

Our Contributions: We propose a resource-efficient traffic normalization scheme that protects contextual information under colluding eavesdroppers. Our scheme achieves perfect privacy while the number of bogus traffic sources is reduced. We map the problem of reducing the bogus traffic sources to the problem of partitioning the WSN into minimum connected dominating sets (MCDSs). Due to the problem complexity, we propose a distributed heuristic algorithm that approximates the WSN partition to MCDSs. We further propose a schedule assignment scheme that reduces packet delay by loosely coordinating transmissions among neighboring sensors.

The remainder of the paper is organized as follows. In Section 2, we present related work. In Section 3, we state our model assumptions. Section 4 presents our traffic normalization scheme. In Section 5, we conduct a performance evaluation and in Section 6, we conclude.

2. RELATED WORK

The problem of hiding contextual information in WSNs has been studied under a local and a global adversary model. Due to space limitations, we focus on the latter model, which is most relevant to our work.

WiSec'13, April 17-19, 2013, Budapest, Hungary.

In [9], the authors proposed two traffic normalization methods based on the injection of bogus traffic; periodic collection and source simulation. In periodic collection, each sensor generates bogus packets at a constant rate. To transmit real data, sensors simply substitute dummy packets with real ones. This method prevents colluding eavesdroppers from determining the source of real traffic, the path to the sink, and the sink location, at the expense of significant communication overhead. Our methods achieve the same level of privacy at a considerably lower communication overhead. Source simulation reduces the communication overhead by selecting a subset of sensors as bogus sources, that are chosen to simulate the expected distribution of real events. However, the event distribution must be known apriori.

In [4], the authors proposed a traffic normalization scheme that propagates dummy packets in a probabilistic fashion. A sensor that overhears the transmission of a real packet, forwards a dummy packet to its neighbors with some probability p. The packet is probabilistically flooded in a radius of K hops from the bogus source. Under a global adversary, if an eavesdropper happens to be close to the source or the sink, their location can be inferred.

Besides their overhead, traffic normalization techniques incur unavoidable delay. This is because transmissions of real packets are delayed to conform to predefined transmission patterns. The authors in [12] reduced packet delay by rushing the transmissions of real packets while delaying the transmissions of follow-up dummy packets so that the long-scale traffic statistics are maintained. This approach is not effective when multiple packets need to be transmitted by the same sensor. Moreover, the authors of [1] proved that statistical analysis of the occurred short-long transmission patterns can be used to identify real packets. To address this vulnerability, the authors proposed the generation of fake short-long patterns by introducing dummy events following packets related to real events.

In [11], the number of bogus traffic sources was reduced by constructing a minimum connected dominating set (MCDS) that covers the deployment area. Only the sensors that belong to the MCDS transmit bogus traffic. Sensors that are not part of the MCDS, regulate their transmissions in order to conform to the statistical traffic properties observed by an eavesdropper. Since the eavesdropper's location is unknown, the set of possible eavesdropped rates is inferred via geometric analysis. The scheme in [11] does not address the case of eavesdropper collusion. The method that we present in our present work provides perfect privacy, even if eavesdroppers collude and can eavesdrop on all network communications.

3. SYSTEM AND ADVERSARY MODELS

System Model: We consider a WSN consisting of a set of sensors $\mathcal{V}$. The WSN is organized as a multi-hop mesh topology, which is defined by the sensor communication range and the sensor positions. Sensors are synchronized to a common time reference. Packets are assumed to be re-encrypted on a per-hop basis to prevent eavesdroppers from identifying a packet relayed over multiple hops [8]. Re-encryption is applied to all packet identifiers such as headers at the MAC layer and the payload. Sensors are pre-loaded with secrets that can be used to establish cryptographic keys. Finally, contention management protocols are assumed to conform to the traffic rate assigned to each sensor.

Adversary Model: We assume an unknown number of colluding eavesdroppers to be deployed at unknown locations within the WSN. The set of eavesdroppers observes communication attributes of interest such as the packet sizes, inter-packet times, identity of transmitting nodes (obtained through the unencrypted header fields, or through signal processing techniques). These observations are collectively processed by a central coordinator to extract contextual information. Because the number and positions of the eavesdroppers are unknown, any portion of the WSN communications could be intercepted. In the extreme case, eavesdroppers are able to intercept all packets transmitted in the WSN. This global adversary model is realistic when eavesdropping devices are cheap sensors with similar capabilities to legitimate sensors [4, 9, 12]. Finally, the adversary does not launch active attacks (e.g., jamming, packet modification and injection attacks), or compromise and control any of the sensors in $\mathcal{V}$.

4. RESOURCE-EFFICIENT TRAFFIC NORMALIZATION

In this section, we develop a resource-efficient traffic normalization scheme to prevent the leakage of contextual information. Our scheme consists of two phases: network partition and schedule assignment. First, we motivate our design.

4.1 Design Motivation

Our design is motivated by the excessive communication overhead of state-of-the-art traffic normalization methods. Since the eavesdroppers' locations are unknown, prior methods hide contextual information by normalizing the transmission profiles of all sensors [4, 9, 12]. Moreover, to hide the route to the sink and the sink's location, transmissions between neighboring nodes remain uncoordinated. Lack of coordination can lead to the accumulation of packet delay on a per-hop basis.

In our design, we represent the WSN as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of sensors, and $\mathcal{E}$ the links between them. Set $\mathcal{V}$ is partitioned into disjoint subsets, denoted by $\{\mathcal{D}_1, \mathcal{D}_2, \ldots \mathcal{D}_z\}$. Only one subset is active at any time and active subsets are periodically rotated in a round-robin fashion. Sensors of an active subset are responsible for normalizing the eavesdropped traffic pattern and relaying real packets to their respective destinations. The partition of $\mathcal{V}$ is designed to form special types of subgraphs that satisfy the following principles: (a) all sensors can transmit real traffic without altering their transmission profile; (b) a subset can deliver a packet to any destination; and (c) the number of bogus traffic sources is minimized. Because only a subset of sensors is active at any given time, the communication overhead is drastically reduced.

To decrease the delay in forwarding real packets, we loosely coordinate the sensor transmissions within each $\mathcal{D}_j$, such that the traffic patterns observed by any number of colluding eavesdroppers remains unchanged. We now describe the two phases in detail.

4.2 Phase I: Network Partition

In the first phase, we partition $\mathcal{V}$ to subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_z\}$. Every subset is active for a fixed time interval. The active subsets are periodically rotated in a round-robin fashion. Sensors of an active subset transmit dummy packets according to a pre-assigned distribution. A sensor with real packets

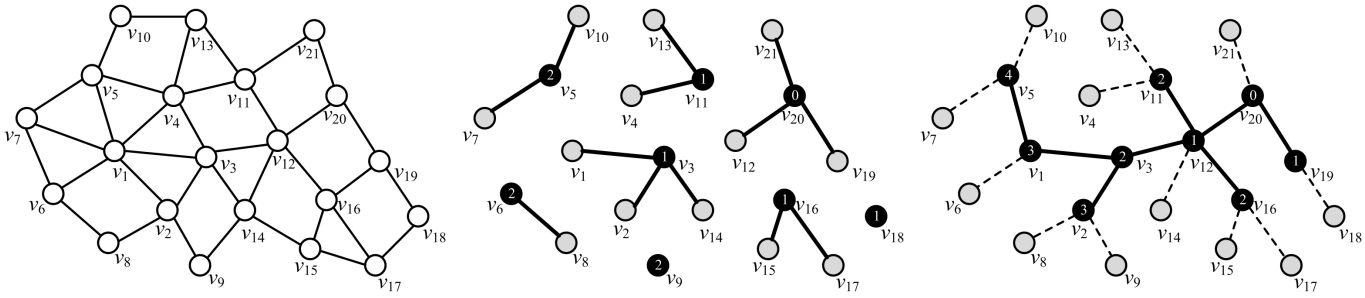


Figure 1: (a) A graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ representing the WSN, (b) a DS generated during Stage 1, (c) an MCDS approximation generated during Stage 2.

for transmission conforms to this distribution by replacing dummy packets with real ones. Thus, transmission of real packets does not alter the traffic patterns observed by eavesdroppers. We reduce the problem of partitioning $\mathcal{V}$, to the problem of *finding disjoint minimum connected dominating sets (MCDSs) that span* $\mathcal{V}$. We now define the MCDS [7].

Minimum Connected Dominating Set: For a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, a subset $\mathcal{D} \subseteq \mathcal{V}$ is a *dominating set (DS)* if any vertex $u \in \mathcal{V}$ either belongs to $\mathcal{D}$, or is adjacent (within one hop) to some vertex in $\mathcal{D}$. If $\mathcal{D}$ induces a connected subgraph on $\mathcal{G}$, then $\mathcal{D}$ is a *connected dominating set (CDS)*. If $\mathcal{D}$ has the smallest possible cardinality, it forms a minimum connected dominating set (MCDS).

The partition of $\mathcal{V}$ into disjoint MCDSs satisfies properties (a)-(c). Property (a) is satisfied, as the set of MCDSs spans $\mathcal{V}$. Hence, each sensor belongs to one $\mathcal{D}_j$, and is able to transmit real traffic when $\mathcal{D}_j$ becomes active. By design, the transmission profile of an active sensor is not altered when real traffic substitutes bogus traffic. For property (b), a CDS guarantees that any sensor in $\mathcal{V}$ will be either part of $\mathcal{D}_j$ or within one hop from a sensor in $\mathcal{D}_i$. Moreover $\mathcal{D}_j$ is a connected set. Hence, a real packet transmitted by a sensor in $\mathcal{D}_j$ can be forwarded to any sensor in $\mathcal{V}$ using only $\mathcal{D}_j$. Finally, property (c) is satisfied by definition, as an MCDS minimizes the number of active sensors.

A partition of $\mathcal{V}$ into disjoint MCDSs does not always exist for arbitrary graph topologies. The number of disjoint MCDSs is bounded by the minimum vertex-cut size of $\mathcal{G}$.Moreover, determining a single MCDS for arbitrary topologies is known to be an NP-complete problem [5]. In the absence of an MCDS partition guarantee and of a polynomial time algorithm for finding an MCDS, we relax the MCDS partition requirement to allow sensors to be part of more than one MCDSs. We denote the frequency of appearance of a sensor v to any of the z MCDSs as $f(v)$. We aim at finding a set of MCDSs that covers $\mathcal{V}$ and balances between the frequency of appearance, number of MCDSs, and MCDS size. We propose a distributed solution inspired by the heuristic MCDS construction algorithm developed in [3]. Our algorithm computes a set of CDSs $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_z\}$, that approximate the partition of $\mathcal{V}$ to MCDSs. We note that the computation and communication overhead for partitioning $\mathcal{V}$ to z CDS is only incurred once during the network initialization. The steps of our algorithm are as follows.

Algorithm 1: MCDS approximation– We generate a $\mathcal{D}_j$ in two stages. We first obtain a DS, and later we expand the DS to a connected graph approximating the MCDS.

For a sensor $v \in \mathcal{V}$, let $m(v)$ be a marker, which can take the values WHITE, BLACK, or GRAY. Let $\mathcal{N}_v$ denote the one-hop neighbors of v, $\delta(v) = |\mathcal{N}_v|$ the degree of v, and $\delta^*(v)$ the *effective degree* of v. Parameter $\delta^*(v)$ is defined as the number of WHITE neighbors of v. Let also $r(v)$ be the rank of v, defined as the order that v changed its marker relative to a leader node. Finally, let $b(v)$ denote the number of higher-ranked BLACK neighbors of v and $f(v)$ the frequency of appearance of v in the CDSs generated thus far. All nodes are initialized to $m(v) =$ WHITE, $\delta^*(v) = \delta(v)$, $b(v) = 0$, $f(v) = 0$, and $r(v) = 0$. The marking process that outputs a DS is as follows.

Stage 1: DS generation

Step 1: A randomly chosen leader v starts the process by changing $m(v)$ to BLACK. Node v becomes a "dominator" and broadcasts $m(v) =$ BLACK, $r(v) = 0$, and $f(v) = 0$.
Step 2: A sensor u with $m(u) =$ WHITE receiving $m(v) =$ BLACK from $v \in \mathcal{N}_u$ is dominated by v. Node u sets $m(u) =$ GRAY, $r(u) = r(v)$, and broadcasts $m(u)$ and $r(u)$.
Step 3: A WHITE sensor v getting $m(u) =$ GRAY from $u \in \mathcal{N}_v$, decreases $\delta^*(v)$ by one, updates the rank to $r(v) = r(u)+1$ if $r(v) \leq r(u)$, and broadcasts $\delta^*(v)$, $r(v)$, and $f(v)$.
Step 4: A sensor v changes $m(v)$ to BLACK, if

$$v = \arg\max_{u \in \mathcal{N}_v \cup \{v\}} \left\{ \frac{\delta^*(u)}{\delta^*_{\max}} \times \frac{1}{f(u)+1} \right\},$$

where $\delta^*_{\max} = \max_{u \in \mathcal{N}_v \cup \{v\}} \delta^*(u)$. Node v becomes a "dominator" and broadcasts its new marker value and rank.
Step 5: After receiving the transmission of a BLACK node, a sensor v updates the value of $b(v)$.
Step 6: The marking process is repeated until no sensors are marked as WHITE (i.e., $\delta^*(v) = 0, \forall v \in \mathcal{V}$).

With the termination of Stage 1, all nodes are marked either as BLACK or GRAY, with each GRAY node dominated by a BLACK one. Therefore, the set of BLACK sensors forms a DS. Figure 1(b) shows the DS generated for the graph of Figure 1(a). Since initially $f(v) = 0$ for all sensors, the marking process depends only on $\delta^*(v)$. In our example, v_{20} becomes the leader and broadcasts $m(v)$=BLACK and $r(v) = 0$. Nodes v_{12}, v_{19}, and v_{21} become GRAY and set their rank to zero. In the next iteration, v_3, v_{11}, v_{16}, and v_{18} are added to the DS and change their rank to one. Finally, v_5, v_6, and v_9 are added to the DS and set their rank to two. The network is now partitioned to a set of star subgraphs, where each star consists of a set of GRAY nodes dominated by a BLACK node. The rank of each star increases with its "distance" from the leader node. In Stage 2, we approximate

an MCDS by selecting GRAY nodes that connect the stars. The process is as follows.

Stage 2: Approximation of the MCDS

Step 1: Every GRAY node v broadcasts $b(v)$.
Step 2: Leader node v selects GRAY nodes $u \in \mathcal{N}_v$ with

$$u = \arg\max_{\{\mathcal{N}_v, r(v)=r(u)\}} \left\{ \frac{b(u)}{b_{\max}} \times \frac{1}{f(u)+1} \right\},$$

where $b_{\max} = \max_{\{u \in \mathcal{N}_v, r(v)=r(u)\}} b(u)$ and $b(u), b_{max} > 0$. Node u changes its marker to BLACK and its rank $r(u) = r(v) + 1$. Ties are broken arbitrarily.
Step 3: A node $w \in \mathcal{N}_u$ with $m(w) =$ BLACK and $r(w) = r(u)$ becomes dominated by u. Dominated nodes change their rank to $r(w) = r(u) + 1$ and broadcast their new rank. Any GRAY node $v \in \mathcal{N}_w$ overhearing a message from w updates $b(v) = b(v) - 1$ and changes its rank to $r(w)$.
Step 4: A GRAY node v overhearing a rank update message from a BLACK node u with rank $r(u) < r(v)$ changes its dominating node to u and broadcasts $r(u)$ and $b(v)$.
Step 5: The process is iteratively repeated until all GRAY nodes have $b(v) = 0$.
Step 6: If a BLACK node does not dominate at least one other node it changes its marker to GRAY.

At the end of Stage 2, every GRAY node has $b(v) = 0$, i.e., all BLACK nodes of Stage 1 are dominated. Moreover, BLACK nodes are dominated by GRAY nodes of lower rank. Since the process is initiated by the leader node, every BLACK node dominated by the GRAY node gets connected to the leader. This process terminates when all BLACK nodes are dominated. Thus, the resulting subgraph is connected. That is, the set $\mathcal{D} = \{v : m(v) = \text{BLACK}, v \in \mathcal{V}\}$ forms a CDS.

Figure 1(c) depicts the CDS generated after Stage 2. In Step 1, the leader node v_{20} selects GRAY node v_{12} ($b(v_{12}) > b(v_{19}), b(v_{21})$) to connect to the star subgraphs dominated by v_3, v_{11}, and v_{16}. In Step 2, v_{12} becomes BLACK and broadcasts $m(v_{12})$ and $r(v_{12}) = 1$. In Step 3, nodes v_3, v_{11}, and v_{16} change their rank to two and broadcast their new rank. Nodes v_{21} and v_{14} change $b(v_{21}) = b(v_{14}) = 0$ and broadcast their new values. Moreover, v_{14} is now dominated by v_{12} since v_{12} has a lower rank than v_3. In further iterations, nodes v_1 and v_2 change to BLACK to connect v_5, v_6 and v_9, respectively and produce a CDS. In Step 6, the CDS is pruned to eliminate the leaf BLACK nodes v_6, v_9, and v_{18}.

In the last stage, the CDS generation process is repeated to produce another CDS for the partition of $\mathcal{V}$.

Stage 3: CDS Update

Step 1: Increment $f(v)$ by one unit for all nodes in $\mathcal{D}_j$.
Step 2: Repeat Stages 1 and 2 until $f(v) > 0,\ \forall v \in \mathcal{V}$.

In Stages 1 and 2, a sensor v is added to the CDS according to metrics,

$$\frac{\delta^*(v)}{\delta^*_{\max}} \times \frac{1}{f(v)+1} \text{ and } \frac{b(v)}{b_{\max}} \times \frac{1}{f(v)+1},$$

respectively. These metrics are designed to balance between the CDS size and the number of CDSs. By maximizing $\frac{\delta^*(v)}{\delta^*_{\max}}$, we minimize the CDS size in a greedy fashion. Nodes that dominate the maximum fraction of their neighbors are added to the DS. Similarly by maximizing $\frac{b(v)}{b_{\max}}$, nodes that connect the largest fraction of star subgraphs are added to the CDS. On the other hand, $\frac{1}{f(v)+1}$ favors the selection of nodes that have not been previously included in any CDS. This metric reduces the number of CDSs needed to span $\mathcal{V}$.

The size of each CDSs generated by Algorithm 1 approximates the minimum DS by a factor of eight. Due to space limitations, we provide an inform proof of this claim. We first note that the DSs $\mathcal{D}_j$ generated in Stage 1 are minimal. That is, if a node $v \in \mathcal{D}_j$ changes its color from BLACK to WHITE or GRAY, $\mathcal{D}_j$ no longer forms a DS. This is due to the fact that in Stage 1, a BLACK node only has GRAY neighbors. Hence, a BLACK node that changed its color to GRAY will not be dominated by any other BLACK node. In [2], the authors proved that a minimal DS approximates the minimum DS with an approximation factor of four. In Stage 2, GRAY nodes change their color to BLACK to connect BLACK nodes that belong to the DS. In the worst case scenario, for each BLACK node of the DS, one GRAY node must turn BLACK to connect it to a BLACK node of lower rank (line network topology). Thus, the size of each CDS is at most twice the size of the DS generated in Stage 1. Therefore, the CDS generated by Algorithm 1 is upper-bounded by a factor of eight times the size of the minimum DS.

4.3 Phase II: Schedule Assignment

In the section, we propose the *Deterministic Assignment Scheme (DAS)* for reducing the end-to-end delay of real packets under a fixed communication overhead budget.

In our scheme, time is divided into intervals $I_1, I_2, \ldots$ of length T. Only one CDS is active at a given interval. We assume T is sufficiently long to accommodate a number of packets according to the given packet rate, and resolve any contention between active sensors within the same collision domain. The CDSs obtained by Algorithm 1 are periodically activated in a round-robin fashion, allowing all sensors to transmit real data. A CDS $\mathcal{D}_j$ is active in interval I_k, if $j = (k \mod z) + 1$. Sensors of an active $\mathcal{D}_j$ either transmit dummy packets, or replace dummy packets with real ones.

4.3.1 Deterministic Assignment Scheme (DAS)

When sensor transmissions are uncoordinated, the packet relay operation during one interval can be blocked if the next hop completes its transmissions prior to the previous hop. This can be illustrated in the CDS of Figure 1(c). Suppose v_3 wants to send a packet to v_{20} (sink). Assume the each sensor randomly selects to transmit one packet within I_k. If the transmission of v_{12} precedes that of v_3, a real packet p will be relayed one time (from v_3 to v_{12}) during I_k. On the other hand, if v_{12}'s transmission follows the transmission from v_3, p will be relayed twice during I_k, delivering p at v_{20}.

In DAS, transmissions are coordinated to maximize the number of relay operations per I_k. We label an active CDS $\mathcal{D}_j$ as a tree rooted at the sink s. Packets from any sensor in $\mathcal{D}_j$ are delivered to s using shortest path routing on the tree. Hence, a packet originating from a sensor v located at depth $d(v)$ requires $(d(v) - 1)$ relay operations until it is delivered to s. We divide each I_k to subintervals $\{I_k^1, I_k^2, \ldots, I_k^\ell\}$ of duration $\frac{T}{\ell}$, where ℓ is the height of the tree. A sensor v at depth $d(v)$ is scheduled to transmit during subinterval $I_k^{\ell - d(v)+1}$. Formally, DAS implements the following steps.

Algorithm 2: Deterministic Assignment Scheme (DAS)

Step 1: $\mathcal{D}_j$ is labeled as a tree rooted at the sink s.
Step 2: A sensor v located at depth $d(v)$ is labeled with $id_v = (d(v) \mod \ell) + 1$, where ℓ is the height of the tree.

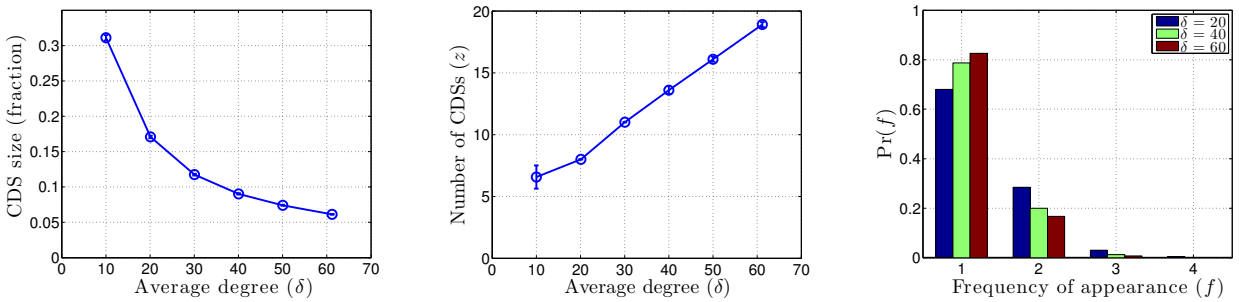


Figure 2: (a) Average CDS size, normalized over the WSN size, as a function of δ, (b) average number of CDSs needed to span $\mathcal{V}$ as a function of δ, (c) empirical probability mass function of f.

Step 3: A sensor with id_v is assigned to transmit one packet in subinterval $I_k^{\ell - id_v}$.

For the CDS $\mathcal{D}_j$ shown in Figure 1(c), suppose that v_{20} is the sink. We first label $\mathcal{D}_j$ as a tree rooted at v_{20}. The label id of each sensor is shown within the circle. Interval I_k is divided into four subintervals (the depth of the tree is $\ell = 4$). Sensor v_5 ($id_5 = 4$) is scheduled to transmit at I_k^1, sensors with $id_v = 3$ are scheduled in I_k^2, and so on.

Note that the sink need not be part of every CDS. If s does not belong to a CDS $\mathcal{D}_j$, any sensor $v \in \mathcal{D}_j$ one-hop away from the source can be selected as the tree root. Such sensor is guaranteed to exist due to the CDS property.

Security Analysis: In DAS, the transmission patterns observed by eavesdroppers are decorrelated from the real traffic pattern. Thus DAS does not reveal the location and time of occurrence of an event. For instance, suppose that a sensor $v \in \mathcal{D}_j$ observed an event $\epsilon(loc, t)$ occurred at time t and location loc. When $\mathcal{D}_j$ becomes active, v transmits packets related to ϵ towards s. Each sensor on the path from v to s will transmit the packets originated by v at random within the designated subintervals according to DAS. These transmissions will not alter the transmission profile observed by any set of colluding eavesdroppers, as real packets substitute dummy ones. Moreover, since hop-by-hop re-encryption is applied at the link layer, copies of the same packet traversing multiple hops remain indistinguishable. Because the adversary cannot distinguish real packets from dummy ones and the transmission pattern is decorrelated from the event pattern, $\epsilon(loc, t)$ is unobservable. However, DAS reveals the sink's location. This is because the sink is the only sensor transmitting during subinterval I_k^1 (all other sensors have an id larger than 1). Hence, DAS may only be adopted when the sink's location must not remain secret.

When the sink's location must be concealed, we use a mechanism that trades communication efficiency for privacy. We label the set $\mathcal{D}_j$ as a tree rooted at a randomly chosen node v. As in the case of DAS, sensor $u \in \mathcal{D}_j$ transmits according to its depth in the tree. To guarantee the delivery of a packet from any sensor to the sink (which differs from the tree root), we divide interval I_k into 2ℓ subintervals. Each sensor $v \in \mathcal{D}_j$ is assigned to transmit one packet in subintervals $I_k^{\ell - id_v}$ and $I_k^{\ell + id_v}$. Based on this schedule, a real packet originating from any sensor v, will reach the randomly selected root by subinterval I_k^{ℓ}. The real packet will continue its propagation to the rest of the sensors of the tree during subintervals $I_k^{\ell+1}$ to $I_k^{2\ell}$. This mechanism implements a form of flooding, restricted to the sensors of the CDS. Because the tree root is randomly selected, the transmission schedule cannot be used to infer the sink's location.

5. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our scheme in terms of communication overhead and packet delay. Our simulations were developed using MATLAB 2012. The simulation results are based on 10 independent runs.

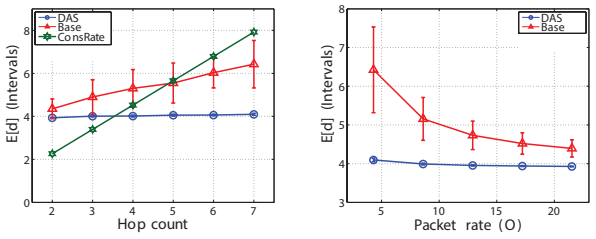
5.1 Generation of an MCDS Partition

In this set of experiments, we studied the performance of Algorithm 1 in terms of (a) the average fraction of sensors that belong in a CDS; (b) the number of CDSs needed to span $\mathcal{V}$; and (c) the probability mass function (pmf) of the frequency of appearance. We randomly deployed a WSN within an area of $1,000$ m $\times 1,000$ m and varied the average node degree δ (by increasing the number of sensors). Sensor locations were randomly drawn from a uniform distribution to generate random topologies. We then applied Algorithm 1 and obtained the set of CDSs $\{\mathcal{D}_1, \ldots, \mathcal{D}_z\}$ that span $\mathcal{V}$.

Figure 2(a) shows the average fraction of $\mathcal{V}$ that belongs to a CDS as a function of δ. Confidence intervals of 95% are also shown. The CDS size indicates the energy savings compared to prior methods that require all sensors in the WSN to be active at a constant rate [9, 14]. We observe the fraction to be as few as 31% of sensors are active when $\delta = 10$, with less than 7% being active when $\delta = 60$. Figure 2(b) shows the average number of CDSs generated by Algorithm 1 as a function of δ. The value of z is a critical factor for the delay until a CDS that contains the real source becomes active. We observe an almost linear increase of z with δ. We note that z implements a tradeoff between the delay and the communication overhead. A partition of the WSN to fewer CDSs increases the size of each CDS and consequently the number of active sensors. However, less time is required to rotate through each of the CDSs. In Figure 2(c), we show the empirical probability mass function of the frequency of appearance f, which is a measure of the "quality" of the partition of $\mathcal{V}$. We observe that more that 67% of sensors are part of only one CDS, while 95% of the sensors have an f less than four. This indicates that Algorithm 1 favors the creation of CDSs that are disjoint to a large degree, reducing the per-sensor communication overhead.

5.1.1 Communication Overhead and Delay

In the second set of experiments, we compared the performance of DAS with the case where sensor transmissions are



(a) (b)

Figure 3: (a) Average delay as a function of the hop count to the sink, (b) average delay as a function of the packet rate.

uncoordinated (referred to as *Base*) and the schemes in [9,12] (referred to as *ConsRate*). In *ConsRate*, perfect contextual privacy is achieved by fixing the packet rate of every sensor. To provide a fair comparison, every scheme was considered under a fixed communication overhead budget. This budget was defined by the number of packets transmitted by all nodes in the network per time interval.

We first considered the end-to-end delay for real packets. Figure 3(a) shows the average end-to-end delay $E[d]$ as a function of the hop count to the sink. The delay is measured in number of intervals until a packet is delivered to the destination. The CDS was rotated per interval. We observe that DAS achieves a constant packet delay irrespective of the hop count to the sink. This is because a real packet always reaches the sink at the end of the interval when the CDS containing the real source becomes active. DAS outperforms the other schemes for a hop count larger than three hops. For shorter hop counts *ConsRate* incurred the lowest delay. This is because for short path lengths, the fixed delay until the corresponding CDS becomes active dominates the overall packet delay. We further observe that DAS introduces a significantly lower delay than the *Base* scheme. Moreover, *Base* has the highest delay variance due to the uncoordinated nature of the real packet relay operation.

We also studied the delay reduction gained by DAS due to the loose coordination of packet transmissions as a function of the average packet rate at each sensor. In Figure 3(b), we compare the packet delay of DAS with the *Base* case. DAS has a fixed delay equal to the CDS rotation delay. On the other hand, in the *Base* scheme the delay decreases with the packet rate. This is primarily due to the reduction of the forwarding delay once a real packet has been transmitted. However, the overall delay is lower-bounded by the delay until a CDS containing the real packet source becomes active.

6. CONCLUSIONS

We addressed the problem of preserving the privacy of contextual information in WSNs under colluding eavesdroppers. We proposed a traffic normalization scheme that significantly reduces the number of bogus traffic sources. This was achieved by partitioning the WSN to a set of CDSs that approximate an MCDS partition. We further reduced the end-to-end packet delay by loosely coordinating the transmissions of sensors within each CDS. We showed that our scheme guarantees the location and time of occurrence privacy of WSN events. Moreover, the end-to-end real packet delay is reduced.

Acknowledgements

This research was supported in part by NSF (under grants CNS-0844111, CNS-1016943, and CNS-1145913) Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES

[1] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran. Statistical framework for source anonymity in sensor networks. In *Proc. of the GLOBECOM Conference*, pages 1–6, 2010.

[2] K. Alzoubi, P. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proc. of the 35th Annual Hawaii International Conference on System Sciences*, pages 3849–3855, 2002.

[3] X. Cheng and D.-Z. Du. Virtual backbone-based routing in multihop ad hoc wireless networks. Technical report, University of Minnesota, 2002.

[4] J. Deng, R. Han, and S. Mishra. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Pervasive and Mobile Computing*, pages 159–186, 2006.

[5] M. Garey and D. Johnson. *Computers and intractability*. Freeman San Francisco, CA, 1979.

[6] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proc. of the ACM MobiSys Conference*, pages 40–53, 2008.

[7] J. Gross and J. Yellen. *Handbook of graph theory*. CRC, 2004.

[8] J. Kong and X. Hong. ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc network. In *Proc. of the MOBIHOC Conference*, pages 291–302, 2003.

[9] K. Mehta, D. Liu, and M. Wright. Location privacy in sensor networks against a global eavesdropper. In *Proc. of the IEEE International Conference on Network Protocols*, pages 314–323, 2007.

[10] A. Proaño and L. Lazos. Packet-hiding methods for preventing selective jamming attacks. *IEEE Transactions on Dependable and Secure Computing*, 9(1):101–114, 2012.

[11] A. Proano and L. Lazos. Hiding contextual information in WSNs. In *Proc. of the IEEE WoWMoM Symposium*, pages 1–6, 2012.

[12] M. Shao, Y. Yang, S. Zhu, and G. Cao. Towards statistically strong source anonymity for sensor networks. In *Proc. of the 27th Conference on Computer Communications*, pages 464–474, 2008.

[13] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders. Short paper: reactive jamming in wireless networks: how realistic is the threat? In *Proc. of the ACM WiSec Conference*, pages 47–52, 2011.

[14] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao. Towards event source unobservability with minimum network traffic in sensor networks. In *Proc. ACM Wisec*, pages 77–88, 2008.