Content-Aware Data Dissemination for Enhancing Privacy and Availability in Wireless Sensor Networks

Miao Xu, Wenyuan Xu, Jason M. O'Kane
Department of Computer Science and Engineering
University of South Carolina, 315 Main Street, Columbia, SC 29208
Email: {xum, wyxu, jokane}@cse.sc.edu

***Abstract*—Wireless sensor networks are vulnerable to various attacks and network dynamics that can breach data privacy and harm data availability. Since those threats cannot be addressed purely by cryptography-based methods, this paper presents a data dissemination scheme that can enhance two goals: data privacy and data availability, leveraging the node location diversity presented in typical wireless sensor networks rather than relying on cryptographic techniques. We demonstrate that the message content is important to quantify the uncertainty associated with data privacy and data availability, and provide content-based definitions utilizing information states. Further, to strike the balance between two conflicting goals in an energy efficient way, we construct a spatial privacy graph based on the locations of network nodes, and use a distributed coloring scheme to ensure that any pairs of nodes whose combined data provide too much information should not send their sensed data to the same storage node. Additionally, sensor nodes selectively send data to multiple storage nodes to achieve higher availability. Our experimental results show that our scheme can achieve better data privacy and a higher level of data availability at smaller energy cost than other baseline data dissemination schemes.**

I. Introduction

Wireless sensor networks (WSNs) are changing the way that we interact with the physical world by providing a low cost method to monitor the surroundings. For example, we have recently witnessed sensing applications that remotely monitor endangered wild animals and track targets. As those WSNs scale in size, the large volume of sensed data and the required energy of collecting them have led to data-centric sensor networks (DCSNs) [1], [2]. In DCSNs, sensed data are stored among a few dedicated storage nodes in the network, and a mobile sink will visit the network occasionally to collect the stored data. Unlike its previous counterpart, the sink-based sensor network where one sink is used to collect and store sensed data, a DCSN is efficient and robust, since it does not require every sensor node to deliver data to the sink that may be far away and may also become a single point of failure.

Once deployed, possibly in a remote environment, DCSNs are typically left unattended with occasional human visits and can create vast quantities of information. As an example, a DCSN can be deployed in a forest for monitoring endangered animals. Sensed data are stored in the network first and are then collected automatically by a forest ranger who makes periodic patrols while carrying a data collector. The characteristic of little physical protection combined with the low cost nature makes DCSNs vulnerable to a wide variety of network dynamics and attacks, including node captures, node compromises, node failures, packet injections, jamming attacks, etc. As a result, an adversary may breach data privacy by acquiring sensitive data stored in the network through node compromises, or may affect data availability by removing data permanently via disabling network nodes. In a habitat monitoring or target tracking DCSN, obtaining the stored data reveals the location information about targets, which may create life-threatening risks. To cope with those threats, we design a data-dissemination scheme that can enhance data privacy and data availability before they are collected by a trusted data center.

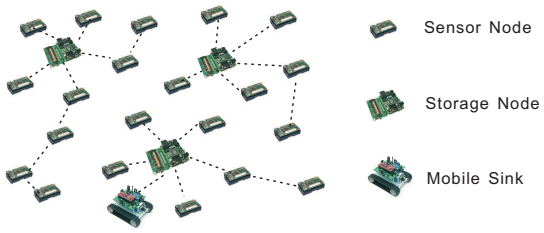


Fig. 1. An illustration of a data-centric sensor network (DCSN).

Many cryptography-based methods [3], [4], [1] are designed to ensure data integrity, confidentiality, and access control for sensor networks. Although those cryptography-based strategies are essential in protecting WSNs against various attacks, they can only partially address the threats against data privacy and data availability. For instance, they cannot cope with information leakage caused by node compromises or communication disturbance caused by jamming attacks. Additionally, most cryptography-based strategies rely on robust key management schemes, which will impose extra storage costs and complicate the network deployment as well as its operations. In this paper, we are interested in whether we can mitigate threats against data privacy and data availability by *non-cryptography-based* methods that only exploit the sensor location diversity exhibited in the typical wireless sensor network. We present a

method that can serve as a complimentary solution to existing cryptography-based methods to enhance data privacy and data availability.

Addressing data privacy issues together with data availability is tricky. To increase data availability against node failure, it is natural to replicate data to many nodes. However, this replication introduces the risk of data privacy leakage due to node compromises. The requirement of energy efficiency further complicates the solution. To strike a balance among these three goals, we construct a graph called the spatial privacy graph (SPG) to guide the data dissemination and validate that our scheme can achieve a higher level of data privacy and data availability at less energy cost compared with other data dissemination schemes. We summarize our contributions as follows:

- We have identified that data privacy and data availability are determined by the uncertainty specified by data contents, e.g., the granularity of locations in the context of location privacy, and we provided a novel definition of data privacy and data availability utilizing *information states* to quantify the uncertainty. Compared to existing privacy definitions, e.g., entropy, our definition requires no prior knowledge.
- To our best knowledge, this is the first work designing a data dissemination scheme with the goal of achieving data privacy and data availability simultaneously. We formulate the problem as a multi-target optimization problem and use distributed graph coloring to drive the data dissemination.
- To solve the problem, we constructed Spatial Privacy Graphs (SPG) by identifying node pairs that compromised in combination can breach data privacy and harm data availability, and designed an SPG-based distributed coloring algorithm that has shown to enhance data privacy and data availability.

The rest of the paper is organized as follows. We begin the paper in Section II by describing the sensor network model and the threat model. Then, we overview the problem of enhancing privacy and availability, and discuss two baseline data dissemination schemes in Section III. We propose our SPG-based data dissemination scheme that utilizes the concept of a SPG in Section IV. In Section V, we evaluate both baseline and our SPG-based data dissemination schemes. Finally, we end the paper with related work in Section VI and concluding remarks in Section VII.

II. Model

In this section, we describe the network model and threat model. We summarize our notations in Table I.

A. Network Model

We focus on a data-centric sensor network that is deployed to track targets or monitor habitats. Specifically, the sensing application first utilizes trusted data collectors to collect messages generated by every sensor, and then derives the location information of the target from the messages. The network consists of sensor nodes, storage nodes, and mobile sinks, as shown in Figure 1.

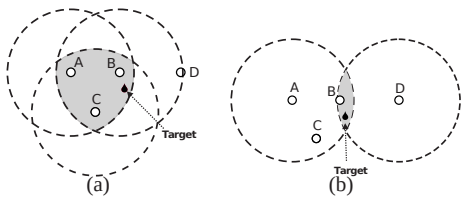


Fig. 2. Illustration that a combination of two potential related nodes provides more valuable information than three nodes which possess similar information.

1) Sensor Nodes: A network of n_n static sensor nodes S_n are deployed through a planar environment W at positions $x_1, x_2, ..., x_{n_n}$ and $S_n = \{x_i\}_{i\in[1...n_n]}$. Each sensor node continually senses its surroundings, and sends an event message to storage nodes whenever it senses an event of interest. Sensor nodes are identical, with the same sensing range r_s and the same communication range r_c. The sensor nodes do not store data, but they always forward data to storage nodes. We avoid letting sensor nodes store data because of their lack of enough memory to store data measured for months or years, and the prohibitive number of nodes from which a mobile sink needs to offload data.

Additionally, we assume the network consists of low cost sensors capable of *coarse sensing*. That is, each sensor is equipped with a long range proximity sensor that can detect the target whenever $\|q(t) - x_{n_i}\| \leq r_s$, where $q(t)$ is the position of a target at time t. This sensing is boolean in the sense that the node knows only whether or not the target has been detected, but no other information. Thus, the reported measurement will be a circle with radius r_s. We assume the r_s is large enough so that capturing one message does not breach the privacy requirement.

Finally, each sensor node is aware of the relative location of its neighbors. Such information can be obtained by wireless localization algorithms [5].

2) Storage Nodes: A collection of n_s storage nodes S_s are deployed across the environment W at position $y_1, y_2, ..., y_{n_s}$, where $n_s \ll n_n$, and $S_s = \{y_i\}_{i\in[1...n_s]}$. Storage nodes have larger size of memory and larger battery capacity. They are in charge of storing data before mobile sinks offload the data. To prevent malicious users from overflowing the storage nodes by injecting faulty packets, each storage node will perform data filtering to sterilize the data. Thus, no matter whether the data are encrypted or not during message deliveries, storage nodes are required to access the plaintext of each packet.

3) Mobile Sinks: From time to time, one or more mobile sinks will visit the network, and they will get close to each storage node to *offload* data. Because of the relatively small number, we assume that mobile sinks are equipped with tamper-proof hardware, or guarded by humans. Thus, mobile sinks cannot be compromised by any adversary or followed by a jammer that may interfere with their communication. In summary, mobile sinks are reliable and trustworthy.

Notation	Explanation	Notation	Explanation	Notation	Explanation
S_n	The set of sensor nodes	n_n	The total number of sensor nodes	x_i	A sensor node, where $i \in \{1, \ldots, n_n\}$
S_s	The set of storage nodes	n_s	The total number of storage nodes	y_i	A storage node, where $i \in \{1, \ldots, n_s\}$
r_s	The sensing radius of sensor nodes	r_c	The communication radius of sensor nodes	p	Duplication probability
$\eta_i(t)$	The I-state of storage node i at time t	$\eta^*(t)$	The master I-state, $\eta^*(t) = \bigcap_{i \in S_s} \eta_i(t)$	$V(\eta(t))$	The area of I-state $\eta(t)$
P	I-state based privacy measure	A	I-state based availability measure	E	Energy cost

TABLE I
FREQUENTLY USED NOTATIONS.

B. Threat Model

In this paper, we consider both unintentional and malicious threats that breach data privacy and harm data availability. We make the following assumptions about the symptoms the adversaries or network dynamics can cause:

Nodes can be compromised. Since both sensor nodes and storage nodes are left in the field unattended and prone to be compromised, we assume both of them are untrustworthy. However, the adversary can only compromise up to g storage nodes, sensor nodes, or any combination of them. As a starting point, we assume $g = 1$ and adversaries are only interested in capturing storage nodes due to the higher payoff of compromising a storage node than a sensor node. When a node is compromised the adversary can obtain all stored data including secret keys and sensed data. Moreover, we assume that adversaries do *not* have a global view of the network and are unaware of all the locations of sensor nodes as well as storage nodes.

Nodes can fail or be jammed. We assume both sensor nodes and storage nodes can fail during the lifetime of the network. They can experience hardware problems, causing permanent data loss, or their communication channel can suffer from severe radio interference, resulting in an inability to receive or send data. In either case, the data that are stored or scheduled to be stored on the affected storage nodes will not be available to mobile sinks.

In summary, data can be leaked to adversaries or can be unavailable to mobile sinks due to various reasons, breaching data privacy and harming data availability.

III. PROBLEM OVERVIEW

In this section, we first motivate the necessity of quantifying privacy and availability based on message contents; then we discuss information states for modeling uncertainty and provide a quantitative definition of data privacy and data availability; finally, we formalize and analyze data dissemination schemes.

A. Privacy Scope

Data privacy of a network includes *content* privacy and *context* privacy [6]. In this study, we focus on content privacy breaches that are caused by node compromises, node failures, or even DoS attacks. We refer readers to other work [7], [6] that copes with preserving context privacy, e.g., where the communication has occurred and who has participated in communication. We note that those two problems are complementary: our *content*-aware data dissemination problem focuses on *which* storage node to deliver while *context*-aware routing problems deal with *how* to deliver data.

B. Motivation for the Privacy and Availability Definition

Preserving privacy is normally considered as the guarantee that data is observable only by those who are supposed to access it. However, such a definition does not capture the fact that privacy is closely linked to its resolution of uncertainty. Taking location privacy for example, we generally do not want to reveal where we are. Here, the definition of *where we are* determines the boundary of the tolerance level of privacy, and it can be quite different in various cases. As an example, Alice might be willing to reveal her location information if the granularity of location is at the level of city, while she is unwilling to reveal her current street address. Similarly, a granularity of no less than 250m may be acceptable for protecting endangered animals, but not less than 25m. Thus, the definition of privacy should quantify the level of information *uncertainty*. Similarly, the goal of data availability is not necessarily to guarantee that all data records are accessible, but to ensure the available data set produces enough information about the target with acceptable level of resolution, i.e., uncertainty.

Before quantifying the information uncertainty, we clarify the relationship between information and messages in sensor networks. Since the message generated by each node only provides a portion of the global location information that the sensing application has, one naive method to quantify the information uncertainty is to count the number of messages. For instance, breaching data privacy can be quantified by the number of messages obtained by adversaries, and data availability can be defined as the number of available messages.

However, with regard to privacy and availability, the *content* of messages is more important than the quantity of messages. Figure 2 provides a simple illustration of the idea in the context of target tracking applications, where the content refers to the location of target. In the figure, nodes A, B, C and D detect the target using their proximity sensors, and each generates a message reporting the possible region of the target as a circle centered at itself. The location information of the target provided by a set of messages is the intersection of corresponding disks. Combining three messages from nodes A, B, and C results in an intersection region much larger than the intersection of nodes A and D's sensing ranges. Thus, leaking three messages does not necessary map to a worse privacy breach than leaking two messages, and the definition

of data privacy and data availability should be *content-aware* rather than counting the messages.

C. Uncertainty and Information States

1) Modeling the Uncertainty: We use the concept of *information states* (I-states) [8] to capture the tolerance level of uncertainty on both privacy and availability associated with a set of messages. I-states are used in robotics for reasoning about uncertainty and explicitly encode the uncertainty about the target. More precisely, we use the term *state* to refer to an instantaneous description of this target at a given time. In target tracking, I-states are the set of *possible states* that are consistent with the measurements provided by sensors, e.g., the possible locations of the target that can incur the measurements, and I-states are calculated according to the content of messages. The main advantage of using the concept of I-states is that no prior knowledge of the target is required but the message contents. In comparison, *entropy* has been used to define privacy [9], [10], but it is only applicable to limited scopes because its calculation requires prior knowledge of the probability distribution for the targets' movements.

Formally, in a network that tracks the motion of a target through a planar environment W using proximity sensors, suppose that prior to some time t_f sensor nodes have measured m samples that map to m messages,

$$\{(O_1, t_1), ..., (O_m, t_m)\}, \tag{1}$$

in which O_i is a circle known to contain the true state, and t_i is a time stamp at which this information was known to be valid. Then a target position $\hat{q}$ is *consistent* with those messages if and only if there exists a continuous trajectory $q : [0, t_f] \to W$ such that

1) $\mathrm{d}q/\mathrm{d}t \leq v_{max}$ for all $t \in [0, t_f]$, where v_{max} is the target's maximum speed;
2) $q(t_i) \in O_i$ for all $i \in [1, m]$;
3) $q(t_f) = \hat{q}$.

The I-state $\eta(t)$ at time t is the *set* of target positions consistent with the messages with the time stamps prior to time t. $V(\eta(t))$ denotes the area of the I-state $\eta(t)$, which quantifies the level of uncertainty. A larger $V(\eta(t))$ means that the target can be anywhere inside a larger area, corresponding to higher level of uncertainty.

Consider the example illustrated in Figure 2 (a), and assume at time $t = 0$ nodes A, B, and C generate three messages. The I-state $\eta(0)$ associated with all three messages is the points inside the intersection of those three disks centered at nodes A, B, and C, respectively; and $V(\eta(t))$ is the area of that intersecting region, denoted by the shaded region in Figure 2 (a).

2) Computing the Information State: To calculate the I-state, we perform iterative updates, maintaining the current I-state and updating it when time passes and when new messages are received. We start with the initial I-state $\eta(0) = W$. Then two kinds of updates are performed throughout the execution:

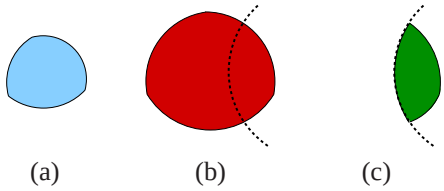


Fig. 3. Computing the I-state. (a) An initial information state. (b) Expansion to account for the passage of time, and intersection with received message disks. (c) The resulting updated I-state.

- When time from t_1 to t_2 passes without any messages being received, we compute $\eta(t_2)$ from $\eta(t_1)$. To accomplish this we perform a Minkowski sum of $\eta(t_1)$ with a ball of radius $(t_2 - t_1)v_{max}$. Informally, this "expands" the I-state to reflect the fact that the state may have changed since the previous message was received. The resulting region is retained as $\eta(t_2)$.
- When a message (O, t) is received, the existing I-state is updated to the correct $\eta(t)$ by intersecting the current I-state with O. This takes the information provided by the message into account.

Figure 3 illustrates each of these updates. Our implementation approximates the curved boundaries of the I-states as polygonal chains.

3) Information States in the Network: For a network with n_s storage nodes, each storage node y_j will calculate its I-state $\eta_j(t)$ based on its received messages. Additionally, there exists a "master" I-state $\eta^*(t)$ derived from all the messages received across all storage nodes, and $\eta^*(t) = \eta_1(t) \cap \cdots \cap \eta_{n_s}(t)$. Thus, there exist $n_s + 1$ I-states in the network in total.

In a normal scenario without any attacks or hardware failures, the mobile sink is able to collect all data stored at each storage node and to obtain $\eta^*(t)$, while in practice some storage nodes may fail and prevent the mobile sink from obtaining $\eta^*(t)$, reducing the amount of information available to the mobile sink. Moreover, it is possible that an adversary compromises one storage node y_j and acquires its I-state $\eta_j(t)$, breaching the network privacy.

D. Evaluation Criteria

We target to design an energy-efficient data dissemination scheme that can enhance privacy and availability. Thus, we define three evaluation metrics.

1) Privacy: Consider the case that the adversary is able to compromise one storage node i. We define the levels of this privacy breach as the size ratio between $\eta_i(t)$[1], which the adversary can access, and $\eta^*(t)$, which is the knowledge of the entire network. This ratio is a measure of the quantity of information that is protected in spite of the compromise. Of course, compromising different storage nodes may lead to a different level of payoff. In light of the fact that security is typically determined by the weakest point in the system, we

[1]Since adversaries do not have the global information of the network, we do not consider the privacy breaches caused by the absence of sensed data at storage nodes, e.g., node A did not detect a target.

define privacy by considering the worst case across all possible storage node compromises:

$$P = 1 - \frac{V(\eta^*(t))}{\min_{i \in S_s} V(\eta_i(t))} \quad (2)$$

for the privacy level at time t. The interpretation of this metric is that when $P = 0$, a single storage node has access to the full knowledge of the network, and privacy cannot be preserved against a compromise of that storage node. Similarly $P = 1$ would indicate "perfect" privacy, but this clearly cannot be achieved, since it would require the network to retain information that is not stored at any of its storage nodes.

2) Availability: Similar to the privacy definition, to define network availability, we consider the area of the I-state available to the entire network, in comparison to the area that is stored at each individual storage node. If a storage node fails, then the knowledge that can be reconstructed from the remaining $n_s - 1$ storage nodes is simply the intersection of their I-states. As a result, we can define availability by considering the worst case across all possible storage node failures:

$$A = \frac{V(\eta^*(t))}{\max_{i \in S_s} V(\bigcap_{j \in S_s - \{i\}} \eta_j(t))} \quad (3)$$

To interpret this metric, observe that if all of the messages are sent only to a single storage node, then we obtain $A = 0$, the worst availability, since the network then has a single point of failure. In contrast, if each message is sent to at least two distinct storage nodes, then $A = 1$, the "perfect" availability, because no single failure can result in data loss. Realistic, energy efficient protocols fall somewhere between these two extremes.

3) Energy: Because the energy available to each wireless sensor node is generally limited by battery capacity, one important objective is to minimize the amount of energy consumed for delivering messages per unit time. Let $E(i)$ denote the number of messages forwarded or generated by the sensor node i between $t = 0$ and $t = T$. The system seeks to keep E as small as possible

$$E = \frac{1}{T} \sum_{i=1}^{n_n} E(i). \quad (4)$$

We note that this energy representation is sufficient to model energy spent both at the sending end and at the receiving end, since we can scale up E by multiplying by a coefficient α. The coefficient α can include the energy consumed both as the sender transmits the message and as its neighbors overhear and process the message.

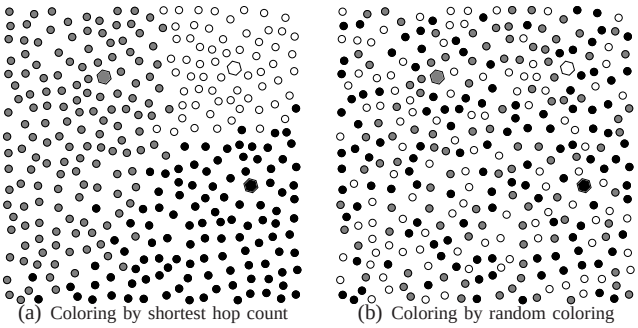
E. Problem Definition

The goal of our data dissemination protocol is to let sensor nodes determine to which storage node they should deliver their observations so that the overall privacy P and availability A are both good while the energy consumption E is small. As such, the data dissemination protocol can be modeled as a color assignment function. We label each storage node with a *unique color ID*, for instance, the same as the storage node ID; and assign colors to each sensor to indicate which storage nodes to deliver its data to. Define the color assignment $\mathcal{C}$ as a function mapping each sensor node x_i to one or multiple storage nodes in S_s, i.e.,

$$\mathcal{C} : S_n \to 2^{S_s},$$

where 2^{S_s} is the power set of S_s. The problem of preserving privacy and availability is equivalent to *finding a color assignment function* $\mathcal{C}$ that maximizes the privacy and availability of the network at the minimum energy cost.

Solving this non-linear multi-objective optimization problem is challenging, since these three evaluation criteria, P, A and E are at least partially in conflict with one another: intuition suggests—and our experiments confirm—that increasing A generally reduces P and increases E. To tackle the problem, we first analyze a few baseline data dissemination technologies to gain insights, and then present our SPG-based data dissemination protocol in Section IV.



(a) Coloring by shortest hop count (b) Coloring by random coloring

Fig. 4. Illustration of shortest path coloring and random coloring.

F. Baseline Data Dissemination

Essentially, we design our data dissemination protocols with the inspiration from secret splitting algorithms [11]. Each sensor is capable of observing a coarse measurement of the target, similar to the concept of small pieces of secret. Storage nodes combine multiple messages, analogous to gaining larger portions of the secret. Finally, the trusted data collector can obtain $\eta^*(t)$ by combining all messages and can pinpoint the location of the target, corresponding to obtaining the secret.

Intuitively, the data dissemination protocol should guide the messages to be distributed across several storage nodes, and thus split the secret evenly among storage nodes. To illustrate this intuition, we analyze two baseline data dissemination protocols.

Shortest path. The shortest path coloring algorithm represents general data dissemination schemes [12] that aim at reducing energy consumption without considering data privacy or data availability. It involves a sensor node choosing the closest storage node to store its data. Figure 4(a) depicts an example of such a coloring scheme with three storage nodes, in which each sensor node transmits to the closest storage node, measured by hop counts in the network, i.e., $\mathcal{C}(x_i) = \arg\min_{y_j \in S_s} \mathsf{h}(x_i, y_j)$, where $\mathsf{h}()$ returns the hop

Scheme	Shortest path	Random color
P	0.30	0.49
A	0.02	0.28
E	36	61

TABLE II
COMPARISON OF THE SHORTEST PATH COLORING AND THE RANDOM COLORING SCHEMES IN A NETWORK OF 3 STORAGES NODES.

count between x_i and y_j. Although such a shortest-hop-count based coloring scheme consumes the smallest amount of energy, it will not provide good privacy and availability. For instance, imagining a target is moving in the white region (upper-right corner), the I-state stored on the white storage node $\eta_w(t)$ equals $\eta^*(t)$. If the white storage node happens to be compromised, the adversary can obtain the same location information about the target as the trusted data collector. Moreover, if the white storage node is unavailable due to hardware failures, then no target movement information will be available. This is the exact situation we want to avoid.

Random coloring. A naive technique to improve the data distribution across the network is to randomly assign each sensor node a color, corresponding to a storage node. That is, the function $\mathcal{C}$ is randomly selected, and only one color is assigned to each sensor node. Figure 4(b) gives an example of random coloring under the same network deployment as Figure 4(a).

Performance comparison. To evaluate the performance of the shortest path and the random coloring schemes, we simulated a network with 325 identical sensor nodes spread across a 2000m by 2000m network field. A single target moved through the field and each sensor node detected the target whenever it was within the sensor's 250m range. The results, which are listed in Table II, confirm that the shortest-path scheme achieves low availability A and privacy P but consumes small amount of energy E. In comparison, the random coloring scheme consumes almost twice the amount of the energy as the shortest-path, but achieves a higher level of data privacy and data availability.

IV. SPG-BASED DATA DISSEMINATION

We have shown that achieving high privacy and availability with minimum energy cost is a tricky multi-objective optimization problem. In this section, we present our SPG-based data dissemination protocol that seeks balance among these objectives.

A. Spatial Privacy Graph

Before defining spatial privacy graph (SPG), we examine the insights obtained from studying our random coloring scheme. Particularly, the random coloring scheme improves the privacy and availability by simply distributing equal numbers of messages to each storage node. However, equal distribution of messages is not sufficient. Recall the example shown in Figure 2, where four nodes A, B, C, and D detect the target. Among all nodes, the combination of A's and D's information states $\eta_A(t) \cap \eta_D(t)$ is more "valuable" compared to

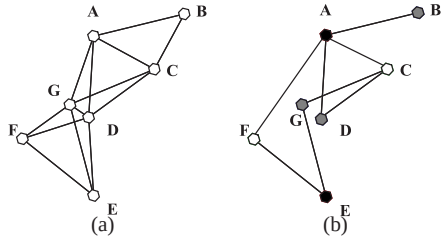


Fig. 5. Illustration of constructing the spatial privacy graph. (a) Communication topology. (b) Spatial privacy graph.

$\eta_A(t) \cap \eta_B(t) \cap \eta_C(t)$. Thus, nodes A and D must transmit their observations to different storage nodes to improve privacy and availability. In contrast, it is relatively harmless for three nodes A, B and C to transmit to the same storage node, because the sensors for these nodes will provide very similar information. This motivates us to construct a spatial privacy graph that identifies those pairs of sensor nodes that in combination can determine the position of the target within a small region.

Formally, we define a spatial privacy graph of a set of sensor nodes S as $G_P = (S, E_P)$ in which a pair of nodes (x_i, x_j) are connected by an edge e_{ij} if and only if they form a *privacy pair*. Given a scalar parameter *privacy factor* a, a pair of nodes is a privacy pair, if their distance $d \in [2r_s - a, 2r_s]$. The intuition is that these privacy pairs are nodes whose sensing regions have small, nonzero intersections. Figure 5 illustrates this process. Figure 5(a) presents a simple network scenario with 7 nodes, where the edges represent communication links. Figure 5(b) depicts the resulting spatial privacy graph, where the edges link privacy pairs. We note that although nodes G and D are within each other's communication range, they are too close to have an overlapped sensing range that is small enough to be considered as a privacy pair. Thus, G and D are not connected in the spatial privacy graph. In this example, we assume $2r_s > r_c$. The distance between node pair (A, F) is larger than their communication range r_c but smaller than $2r_s$. Thus, nodes A and F are not connected in the network topology, but are connected in the spatial privacy graph.

B. Enhancing Privacy via a Distributed Coloring Algorithm

The SPG identifies the privacy pairs that should select different storage nodes to save their data. Thus, to enhance data privacy, each sensor node can determine its storage node by executing a distributed graph coloring scheme. Given an n-vertex SPG with $G_P = (S, E_P)$, the output of the distributed coloring scheme is a colored graph $G_c = (S, E_P, C)$. Without loss of generality, we assign one color to each sensor node, and denote the color assignments C as $C = \{\mathbf{c}_{x_i} | \mathbf{c}_{x_i} = \mathcal{C}(x_i)\}_{\forall x_i \in S}$. Ideally, G_c should satisfy *two* requirements: *valid* and *feasible*. Here, *valid* means that for every edge $e_{ij} \in E_P$, its vertices x_i and x_j have different colors, e.g., $\mathbf{c}_{x_i} \neq \mathbf{c}_{x_j}$, and *feasible* means that the color of every vertex should be one of the storage nodes' colors. A valid and feasible coloring can guide the network to disseminate messages that belong to the same privacy pairs to different storage nodes

and thus achieve high privacy. However, for any SPG and given number of storage nodes, it is not always possible to obtain a valid yet feasible colored graph. For instance, if there are only two storage nodes available to color the SPG shown in Figure 5(b), then it is impossible to obtain a valid coloring among nodes A, C, and D. To address this issue, our distributed coloring algorithm will first generate a *valid* coloring and then adjust those *infeasible* colors into *feasible* colors.

Algorithm walk-through. Our distributed coloring algorithm is motivated by Linial's coloring scheme [13], which starts with a valid colored graph with a large number of colors and then reduces the total number of colors iteratively. However, Linial's coloring scheme is inapplicable to our problem because it does not consider the factor of energy consumption, which is crucial to sensor networks. In comparison, our distributed coloring algorithm is energy efficient.

Our distributed algorithm works in the following way. Prior to coloring sensor nodes, we map each storage node to a unique color numbered from 1 to n_s. Then, each sensor node assigns its color purely based on its neighbors' colors by executing `Distributed_Coloring()` (shown in Algorithm 1) in parallel. Here, we call a pair of nodes *neighbors* if they are connected on the SPG, which is different from the concept of neighbors defined according to communication abilities. Each sensor x_i initializes its color to a unique infeasible one, e.g., adding its own ID I_{x_i} to n_s. As such, we prevent any sensor nodes from pre-assigning itself a feasible color. Then each sensor node participates in iterative coloring updating until no color updates between two consecutive iterations.

At the beginning of each iteration, node x_j announces its current color with its ID I_{x_j} to all its neighbors by broadcasting a message $(I_{x_j}, \mathbf{c}_{x_j})$, where $\mathbf{c}_{x_j}$ is its current color. At the same time, it records its neighbors' current colors $\{\mathbf{c}_{x_i}\}_{x_i \in \mathbf{Nbr}}$. In each iteration, only the sensor nodes that satisfy the following conditions is allowed to update its color:

1) It has not been assigned a feasible color yet.
2) Its color is larger than all its neighbors'.

Function `UpdateColor()` first tries to find a new color that satisfies all conditions listed below.

1) *Feasible*, the new color should be one of the storage nodes' colors, $\mathbf{c}'_{x_j} \in \{1, \ldots, n_s\}$.
2) *Valid*, none of its neighbors has chosen this color, $\mathbf{c}'_{x_j} \notin \{\mathbf{c}_{x_i}\}_{x_i \in \mathbf{Nbr}}$.
3) *Nearest*, among all valid and feasible colors, it chooses the storage node that is separated by the fewest hop counts from itself.

Sometimes it is possible that no feasible and valid color is available, as shown in Figure 5(b). In those cases `UpdateColor()` returns $-|\mathbf{c}_{x_i}|$. We note that the algorithm terminates when none of the nodes can update its color further, and the following Lemma holds. .

Lemma 1. *Algorithm 1 always terminates after $|S|$ iterations and terminates with a valid (but not necessarily feasible) colored graph $G_c = (S, E_P, C)$.*

Algorithm: Distributed_Coloring
Input: **Nbr**: neighbor set
Input: I_o: local sensor ID
$\mathbf{c}_o = I_o + n_s$;
repeat
  Announce($I_o, \mathbf{c}_o$);
  $\{\mathbf{c}_{x_i}\}_{x_i \in \mathbf{Nbr}}$ = ReceiveAnnounce();
  if $\boldsymbol{c}_o > n_s$ **and** $\boldsymbol{c}_o > \max\{\mathbf{c}_{x_i}\}_{x_i \in \mathbf{Nbr}}$ **then**
    $\mathbf{c}_o$ = UpdateColor($\{\mathbf{c}_{x_i}\}_{x_i \in \mathbf{Nbr}}$);
  end
until *NoChange($\boldsymbol{c}_o$)* **and** *NoChange($\{\boldsymbol{c}_{x_i}\}_{x_i \in \mathbf{Nbr}}$)*;

Algorithm 1: The SPG-based distributed coloring algorithm

Proof: We first prove that the algorithm terminates within $|S|$ iterations, then prove the resulting colored graph is valid by induction.

Termination. In each iteration, a node that can update its color must have a color that is larger than n_s. Meanwhile, a node can only update its color either to the number between 1 and n_s, or to its negative node ID. Thus, each node $x_i \in S$ will only update its color at most once. The algorithm terminates when none of nodes can update its color, and the total number of iterations $I \leq |S|$.

Validity. We prove validity by induction on k. Let $G_c^{(0)} = (S, E_P, C^{(0)})$ be the colored graph after initialization, then for all nodes x_i we have $\mathbf{c}_{x_i} = I_{x_i} + n_s$. Since all nodes have unique identifications, $\forall x_i, x_j \in S, \mathbf{c}_{x_i} \neq \mathbf{c}_{x_j}$, $G_c^{(0)}$ is valid.

Assume $G_c^{(k-1)}$ is valid. Let the graph after kth iteration be $G_c^{(k)}$. Since in each iteration only the node that has the largest color in its neighborhood can update its color, we assume w.l.o.g that node x_p updates its color from $\mathbf{c}_{x_p}^{(k-1)}$ to $\mathbf{c}_{x_p}^{(k)}$. According to the color updating condition 2, $\mathbf{c}_{x_p}^{(k)} \neq \mathbf{c}_{x_q}^{(k)}$, for all x_q that are its neighbors. Thus, $G_c^{(k)}$ is valid. ■

When Algorithm 1 produces a valid but infeasible graph, e.g., some sensor nodes have a color that is out of the feasible range $[1, \ldots, n_s]$, the sensor nodes with infeasible color will randomly choose a feasible color regardless of their neighbors' colors. We note that at this step sensor nodes should not select the nearest colors, otherwise it is likely that several of them will choose the same storage node and will reduce the level of privacy.

Algorithm challenges. There are several practical challenges associated with this algorithm.

Loose Synchronization. The correctness of the distributed coloring algorithm holds only if at most one node in its neighborhood updates its color in each iteration. Such a condition can be guaranteed only if every node decides whether it should update its color after all color announcements are delivered. Thus, it is important to let every node have a loosely synchronized clock and to let the color announcements reach its neighbors. For synchronization, one can use TPSN (Timing-sync Protocol for Sensor Networks) [14], a light-weight syn-

chronization protocol. For coloring announcement, we use TTL (time to live) to control the flooding range. The neighbors with regard to the SPG are not communication neighbors. Thus, the coloring announcement has to be broadcast beyond a 1-hop neighborhood. In cases where the communication range r_c equals the sensing range r_s, the privacy pair can be located up to $2r_s$ apart and we set TTL = $2r_s/r_c = 2$.

Reducing energy through on-demand, incremental coloring. Energy-efficiency is one of the main concerns when designing algorithms for sensor networks. Our SPG-based coloring algorithm is energy efficient in the sense that each node always chooses a valid color of the storage node closest to it, and it converges in at most $|S|$ steps. Additionally, we adopt the following rules to further reduce the energy consumption: (1) *Construct the SPG on-demand.* In a tracking sensor network, a few nodes will detect the target; we call those nodes *hot nodes* S_{hot}. Instead of constructing an SPG across the whole network, only hot nodes will participate in constructing the SPG by broadcasting control messages locally. (2) *Incremental coloring.* That is to incrementally update the SPG as the target moves continuously.

The *incremental* coloring algorithm works in the following manner. When the target moves to location L_1 initially, all hot nodes $S_{hot}(L_1)$ will color themselves using Algorithm 1. In the next time window, the target moves to another location L_2, and the $S_{hot}(L_2)$ will intersect with $S_{hot}(L_1)$. The nodes that belong to the intersection $S_{hot}(L_2) \cap S_{hot}(L_1)$ keep their color unchanged, and the nodes that are part of the set $S_{hot}(L_2) - S_{hot}(L_1)$ select their colors. As such, the colors of $S_{hot}(L_2) \cap S_{hot}(L_1)$ can be treated as prior knowledge, and only nodes in the set $S_{hot}(L_2) - S_{hot}(L_1)$ need to announce and update their colors iteratively. We note that this incremental coloring is especially beneficial in reducing energy cost when the target moves at a low speed.

C. Enhancing Availability via Message Replication

In a non-failure scenario, the mobile sinks can derive $\eta^*(t)$ by acquiring data from every storage node. However, the data stored on storage nodes may be unavailable due to hardware failure or jamming attacks. The goal of maintaining high data availability is to ensure that the intersection of the information state of available storage nodes, $\bigcap_{i \in S_s} \eta_i(t)$, is close to $\eta^*(t)$. A natural way to improve high availability involves replication, e.g., let a sensor node deliver a copy of the data to another storage node. However, naive duplication will increase the energy cost. To replicate efficiently, we ask three questions: (1) who should duplicate its messages, (2) how, and (3) where should the duplicated messages go?

Who? Only privacy pairs shall duplicate their messages. This heuristic can be illustrated by the example in Figure 6, which consists of two privacy pairs, (B, D) and (B, E), and isolated nodes A and C. The nodes that do not form privacy pairs with any hot nodes are usually located in the center of hot nodes. Their intersection (denoted by the light grey shadow) is typically larger than the interaction of privacy pairs, and thus is less valuable towards increasing availability. Letting privacy pairs duplicate messages allows us to spend energy on the most valuable messages.

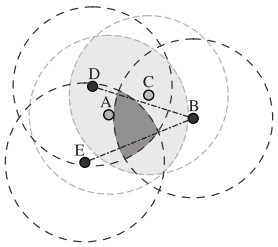


Fig. 6. SPG and information redundancy. Node B, D and E form privacy pairs, and their intersected sensing area is contained by the intersection of A's and C's sensing regions.

How? Availability and privacy are conflicting objectives. Thus we adjust the duplication probability p to balance between two goals. Each node that is part of privacy pairs will replicate messages with probability p. Particularly, in each data reporting period, a node generates a random number in the range of $[0, 1]$. Only if the random number is smaller than p will it send a replicated message to a second storage node. Setting $p = 0$ gives privacy higher priority while assigning $p = 1$ favors availability.

Where? To avoid the situation that the duplicated messages from the same region are always delivered to the same storage node, the privacy pairs will randomly choose a second storage node to deliver their duplicated messages.

V. Experiment Validation

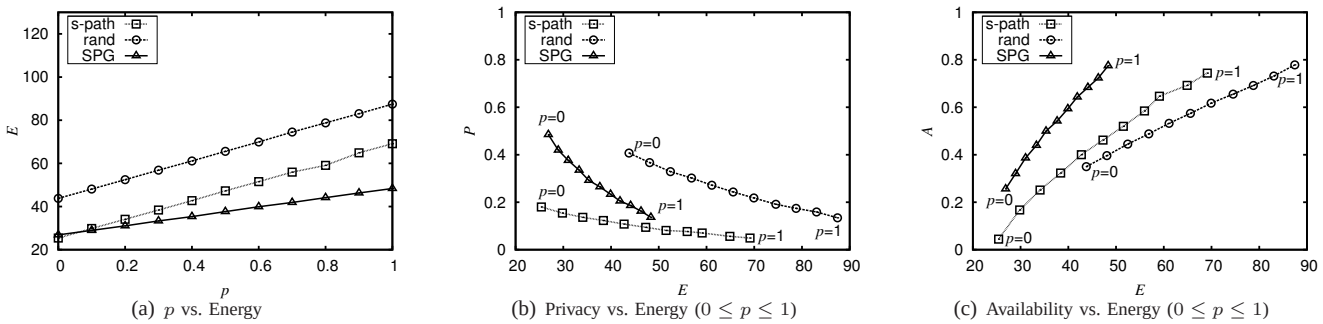
A. Simulation Methodology

We have implemented the SPG-based data dissemination algorithm with C++. We simulated a sensor network deployed in a 2000m-by-2000m region with $r_s = r_c = 250$m, and a target moved randomly throughout the network region at a speed of 25m/s. We studied all three data dissemination strategies: the shortest path, random coloring, and our SPG-based algorithm. For the SPG-based algorithm, we set the privacy factor a to 15m and measured the energy cost for both constructing SPG and delivering data. To capture the statistical characteristics, we evaluated P, A, and E by running our experiments 10 rounds and each round lasted for 1000 seconds with a 1 second sensing interval .

B. Experiment Results

We performed two sets of experiments to study the impact of p and the number of storage nodes n_s, respectively.

1) Impact of p: We first compared the performance of three algorithms in the scenarios of 200 sensor nodes and 3 storage nodes when varying p from 0 to 1. The results are depicted in Figure 7, from which we observed that the availability of all three algorithms improves with p increasing but at the cost of less privacy and higher energy cost. Compared with the other two algorithms, the energy cost of the SPG-based algorithm grows slower. Interestingly, when p is larger than 0.1, the energy cost of the SPG-based algorithm becomes smaller than



(a) p vs. Energy (b) Privacy vs. Energy ($0 \leq p \leq 1$) (c) Availability vs. Energy ($0 \leq p \leq 1$)

Fig. 7. Comparison between shortest path coloring, random coloring and the SPG-based algorithm with p changing from 0 to 1. $n_n = 200$. $n_s = 3$.

the one of the shortest path scheme. This is because our SPG-based algorithm only allows privacy pairs to duplicate messages instead of all hot nodes.

Figure 7(b) shows P and E for all three algorithms. Note that the point at $(0, 1)$ represents the (unachievable) ideal of perfect privacy with no energy cost. Figure 7(b) shows that the SPG-based algorithm accomplishes higher privacy than the shortest path scheme, which can only achieve a maximum privacy of 0.2. Compared with the random coloring scheme, the SPG-based algorithm can achieve the same level of privacy with less energy cost.

Finally, Figure 7(c) shows that the SPG-based algorithm dominates both the shortest path and random coloring schemes with regard to A and E. That is, at the same energy cost the SPG-based algorithm provides highest availability.

2) Impact of n_s: Besides tuning p to balance between A and P, it is interesting to know what the maximum achievable A is, given the energy budget and the minimum required P. Figure 8(a) and Figure 8(b) show such cases with the requirements of $E \leq 50$ and $P \geq 0.4$. As n_s becomes larger than 4, the SPG-based algorithm outperforms the random coloring schemes, and uses a smaller amount of energy. Moreover, we observe that in Figure 8(a), with the increases of the storage-node number, the availability of SPG-based algorithm increases much faster than the availability of the random coloring algorithm. This confirms our analysis: distributing messages evenly is insufficient; and the content of messages is more important than the number of messages in terms of data uncertainty. We note that the shortest path algorithm cannot achieve the requirements and does not show up in the plots. Similarly, as shown in Figure 8(c) and Figure 8(d), given the requirements of $A \geq 0.6$ and $E \leq 50$, the SPG-based algorithm achieves higher maximum privacy than the shortest path scheme and uses less energy. We note the random coloring scheme cannot find any feasible solution to meet the requirements and does not appear in the plots.

In summary, our SPG-based data dissemination protocol combines the advantages of two baseline dissemination schemes and can achieve better data privacy and a higher level of data availability while consuming less energy.

VI. Related Work

Much attention has been devoted to addressing privacy issues in the context of data mining and databases [15], [16], [17]. A common technique is to perturb the data and to reconstruct distributions at an aggregate level. This type of approach is centralized and cannot be applied to resource-constrained sensor networks.

The problem of providing contextual location privacy in WSNs has been well studied. The primary concern of location privacy in WSNs is to protect the source location [6], [18], [19] and sink location information [7]. To protect the source location against a local adversary, phantom routing [6] uses a random walk before commencing with regular flooding/single-path routing. Later, Mehta et al. [18] and Yang at al. [19] studied the source location privacy problem in the presence of a global adversary who can observe all traffic in the network. Mehta et al. proposed to use hop-by-hop encryption to hide the message flows, and Yang et al. proposed to inject fake messages. Deng et al. [7] proposed randomized routing algorithms and fake message injection to prevent an adversary from locating the network sink based on the observed traffic patterns.

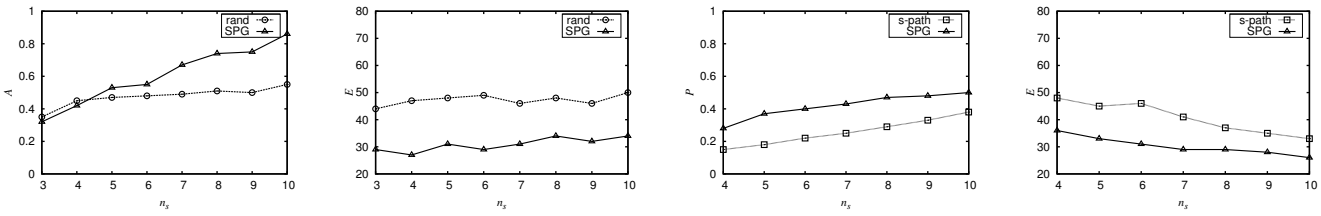
A common design goal of data dissemination protocols [20] in wireless sensor networks is to achieve energy-efficiency. Ugur et al. [20] let data travel down an event dissemination tree based on a schedule to save energy. To address the data privacy issues, Shao et al. [1] designed a data dissemination scheme called pDCS that can provide different levels of data privacy based on different cryptographic keys.

In the areas of constructing storage systems, Gregory et al. [21] and SafeStore [22] have addressed issues of ensuring the system availability and integrity policies in the presence of component failures and malicious attacks.

Unlike prior work, we addressed the problem of data privacy and data availability at the same time using a non-cryptographic method.

VII. Conclusion

Preserving data privacy and data availability in WSNs cannot be achieved purely by cryptographic strategies. In this paper, we proposed an SPG-based data dissemination protocol



(a) Availability (b) Energy (c) Privacy (d) Energy

Fig. 8. Influence of the number of storage nodes when $n_n = 200$. (a) and (b): given the requirements of $P \geq 0.4$ and $E \leq 50$, the maximum achievable A and the corresponding E; (c) and (d): given the requirements of $A \geq 0.6$ and $E \leq 50$, the maximum achievable P and the corresponding E.

that is complimentary to traditional cryptographic techniques and that can enhance data privacy and data availability in sensor networks deployed for target tracking. We argued that data uncertainty is important to quantify data privacy and data availability, and message content is more important than the number of messages with regard to data uncertainty. As such, we provided a content-based definition of data privacy and data availability utilizing information states. To strike a balance between two conflicting objectives, data privacy and data availability, we introduced a graph called spatial privacy graph (SPG) that identifies node pairs whose combined sensed data provide high certainty of the target location, and showed that the task of disseminating data to storage nodes is equivalent to the problem of coloring the SPG.

Our SPG-based data dissemination protocol consists of the following steps: (1) Constructing the SPG among hot nodes (nodes that detect the target) on-demand; (2) coloring the SPG using our energy-efficient distributed coloring algorithm; (3) letting those nodes that provide "valuable" information replicate messages with a probability p. Our experiment results have shown that our SPG-based data dissemination protocol combines the advantages of two baseline dissemination schemes: shortest path routing and random coloring protocols. The SPG-based protocol can achieve better data privacy and a higher level of data availability while consuming less energy than either baseline data dissemination scheme.

ACKNOWLEDGMENTS

This work is partially supported by National Science Foundation Grants CNS-0845671 and IIS-0953503, and by a grant from the University of South Carolina, Office of Research and Economic Development Research Opportunity Program.

REFERENCES

[1] M. Shao, S. Zhu, W. Zhang, G. Cao, and Y. Yang, "pDCS: Security and privacy support for data-centric sensor networks." *IEEE Trans. Mob. Comput.*, vol. 8, no. 8, pp. 1023–1038, 2009.

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of Conference on Mobile Computing and Networks (MobiCOM)*, 2000.

[3] H. Chan and A. Perrig, "Security and privacy in sensor networks," *IEEE Computer*, vol. 36, no. 10, pp. 103–105, 2003.

[4] A. Perrig, R. Szewczyk, D. Tygar, V. Wen, and D. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[5] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *International Conference on Mobile Computing and Networks (MobiCOM)*, 2001, pp. 166–179.

[6] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005.

[7] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," in *Proceedings of Conference on Dependable Systems and Networks (DSN)*, 2004, p. 637.

[8] J. M. O'Kane and W. Xu, "Energy-efficient target tracking with a sensorless robot and a network of unreliable one-bit proximity sensors," in *Proc. IEEE International Conference on Robotics and Automation*, 2009.

[9] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, 2003, pp. 41–53.

[10] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, 2003, pp. 54–68.

[11] W. Trappe and L. Washington, *Introduction to Cryptography with Coding Theory*. Prentice Hall, 2002.

[12] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks," in *Proceedings of the Usenix Symposium on Operating Systems Design and Implementation*, 2002.

[13] N. Linial, "Locality in distributed graph algorithms," *SIAM J. on Computing*, vol. 21, no. 1, pp. 193–201, 1992.

[14] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of conference on Embedded networked sensor systems (SenSys)*, 2003, pp. 138–149.

[15] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. of the ACM SIGMOD Conference on Management of Data*. ACM Press, May 2000, pp. 439–450.

[16] C. K. Liew, U. J. Choi, and C. J. Liew, "A data distortion by probability distribution," *ACM Trans. Database Syst.*, vol. 10, no. 3, pp. 395–411, 1985.

[17] N. Minsky, "Intentional resolution of privacy protection in database systems," *Commun. ACM*, vol. 19, no. 3, pp. 148–159, 1976.

[18] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proceedings of Conference on Network Protocols (ICNP)*, 2007, pp. 314–323.

[19] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proceedings of conference on Wireless network security (WiSec)*, 2008, pp. 77–88.

[20] U. Cetintemel, A. Flinders, and Y. Sun, "Power-efficient data dissemination in wireless sensor networks," in *Proceedings of workshop on Data engineering for wireless and mobile access (MobiDe)*, 2003, pp. 1–8.

[21] G. Ganger, P. Khosla, M. Bakkaloglu, M. Bigrigg, G. Goodson, S. Oguz, V. Pandurangan, C. Soules, J. Strunk, and J. Wylie, "Survivable storage systems," *DARPA Information Survivability Conference and Exposition*, vol. 2, pp. 184–195, 2001.

[22] R. Kotla, L. Alvisi, and M. Dahlin, "Safestore: A durable and practical storage system," in *In USENIX Annual Technical Conference*, 2007, pp. 07–20.