8-29-2005

Quantifying Eavesdropping Vulnerability in Sensor Networks

Madhukar Anand
University of Pennsylvania, anandm@cis.upenn.edu

Zachary G. Ives
University of Pennsylvania, zives@cis.upenn.edu

Insup Lee
University of Pennsylvania, lee@cis.upenn.edu

Postprint version. Copyright ACM, 2005. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *Proceedings of the 2nd International VLDB Workshop on Data Management for Sensor Networks 2005 (DMSN 2005)*, pages 3-9.
Publisher URL: http://doi.acm.org/10.1145/1080885.1080887

Quantifying Eavesdropping Vulnerability in Sensor Networks[*]

Madhukar Anand
Department of Computer and Information Science
University of Pennsylvania
anandm@cis.upenn.edu

Zachary Ives
Department of Computer and Information Science
University of Pennsylvania
zives@cis.upenn.edu

Insup Lee
Department of Computer and Information Science
University of Pennsylvania
lee@cis.upenn.edu

ABSTRACT

With respect to security, sensor networks have a number of considerations that separate them from traditional distributed systems. First, sensor devices are typically vulnerable to physical compromise. Second, they have significant power and processing constraints. Third, the most critical security issue is protecting the (statistically derived) *aggregate* output of the system, even if individual nodes may be compromised. We suggest that these considerations merit a rethinking of traditional security techniques: rather than depending on the resilience of cryptographic techniques, in this paper we develop new techniques to *tolerate* compromised nodes and to even mislead an adversary. We present our initial work on *probabilistically quantifying* the security of sensor network protocols, with respect to sensor data distributions and network topologies. Beginning with a taxonomy of attacks based on an adversary's goals, we focus on how to evaluate the vulnerability of sensor network protocols to eavesdropping. Different topologies and aggregation functions provide different probabilistic guarantees about system security, and make different trade-offs in power and accuracy.

Categories and Subject Descriptors: C.2.0 [Computer-Communication Networks]: Security and Protection

General Terms: Security

Keywords: Wireless Sensor Networks, Eavesdropping, Data Streams, Probability Distribution.

1. INTRODUCTION

As sensor network technology advances, security and privacy concerns will increasingly move to the forefront. Many real-world settings in which sensors might be deployed (e.g., security systems, intelligent buildings, hospitals, automated warehouses) have significant need not only for privacy policies, but mechanisms for enforcing data security and confidentiality.

In the Aspenn (Abstraction-based Sensor Programming Environment from Penn) project, we focus on developing the infrastructure for such rich sensor applications, in which the sensing devices and networks may be heterogeneous (including smart card readers, video cameras, and mobile sensors) and the sensor network may interact with external data sources on the Internet. A major emphasis of our work lies in protecting application data from eavesdroppers and hackers.

With respect to security, the sensor network domain has several important characteristics that differentiate it from traditional distributed systems. First, sensor devices are frequently vulnerable to *physical compromise or local eavesdropping*, as they are embedded within an environment. Second, sensor devices have significant *power and processing constraints*, which often prevent them from running expensive encryption protocols, but which also limit the amount of "damage" they can do to the overall sensor network (e.g., by injecting spurious data or snooping on large volumes of messages). Third, sensor network applications are generally consensus or *aggregation*-based, meaning that compromising one or a few nodes may not significantly affect the overall system.

To this point, security techniques have been adapted for the sensor network domain by reducing the computation requirements of cryptography (generally by pre-distributing keys [18] or reducing the key size [2]) in order to operate under the limited processing capabilities of sensor networks. However, cryptography is not the only means of providing security in a sensor network application — in fact, if an attacker has sufficient resources, cryptographic schemes with small key sizes may provide little protection. Moreover, such techniques do not consider the *system-wide* effects if an attacker compromises a few nodes.

We advocate a different approach, which takes advantage of the fact that any real-world attacker is limited by the properties of the system he or she is attempting to compromise. In this paper we present an initial framework, taxonomy, and methodology for *quantifying* the privacy and security of sensor network applications, under the assumption that some nodes may be compromised, and based on the networks' size, protocols, and computations. Rather than providing all-or-nothing guarantees about privacy or security, our goal is to examine *probabilistic guarantees* with respect to compromise, and to understand and improve existing aggregation strategies with respect to these guarantees. Our focus in this paper is on the problem of eavesdropping, although we are currently generalizing to other types of attacks. Specifically, we make the following contributions:

- We propose a taxonomy of attack models for sensor networks, based on the *goals* of the attacker.

[*]This work was funded in part by NSF grants IIS-0477972 and CCR-0209024 and ARO grants DAAD19-01-1-0473 and W911NF-05-1-0182.

DMSN'05, August 29, 2005, Trondheim, Norway.
Copyright 2005 ACM 1-59593-206-2/05/0008 ...$5.00.

- We propose what we believe to be the first *quantitative* approach to assessing *system-level* confidentiality and security, under the possibility that some nodes are compromised.

- We show how our methods can be used to choose between different protocols and sampling strategies.

- We discuss how cryptographic and non-cryptographic techniques can be used to improve the confidentiality of a sensor network.

The remainder of the paper is organized as follows: in Section 2, we introduce a taxonomy of attacks in sensor networks. In the subsequent section, we develop a model for cost and accuracy in a sensor network. Section 4 discusses how we model an attacker's ability to determine the output of a sensor network, and also her cost. Next, we identify and assess potential means of combating eavesdropping. We discuss related work in Section 6, and in Section 7 we conclude by highlighting avenues for future work.

2. TAXONOMY OF ATTACKER MODELS

By compromising nodes, eavesdropping, or spoofing, an adversary may attempt to violate the security of a sensor network application. In order to evaluate a sensor application's security characteristics, we must first understand the potential goals of the adversary's attack. We define a taxonomy of attack models for sensor networks, based on the goals of the adversary.

1. *Eavesdropping.* Here, the adversary (eavesdropper) aims to determine the aggregate data that is being *output* by the sensor network: it is attempting to see what the system is observing, e.g., to predict how the owner of the sensor network will react. The adversary either listens to messages transmitted by the nodes, or directly compromises those nodes. We further distinguish between two types of eavesdropping:

   (a) *Passive:* The eavesdropper conceals her presence from the sensor nodes and uses only the broadcast medium to eavesdrop on all messages.

   (b) *Active:* The eavesdropper actively attempts to discern information by sending queries to sensors or aggregation points, or by attacking sensor nodes.

2. *Disruption.* The intent of the adversary is to disrupt the sensor application. This can be a combination of two types of techniques:

   (a) *Semantic:* The adversary injects messages, corrupts data, or changes values in order to render the aggregated data corrupt or useless.

   (b) *Physical:* The adversary upsets sensor readings by directly manipulating the environment. For example, generating heat in the vicinity of sensors will result in erroneous values being reported.

3. *Hijacking.* This variation on the disruption model is a case in which the adversary attempts to direct the aggregated output of the sensor application towards a value *of her choosing.* If the adversary gains control of enough sensors, then this attack is the hardest to counter.

Our focus. In this paper, which forms the first step towards addressing the attack models of our taxonomy, we focus strictly on the case of eavesdropping. As stated above, we assume that the adversary's goal is to ascertain the aggregated values output by the network: while subtly different from the alternative definition — attempting to precisely ascertain information about the sensed environment — we believe this is a more likely motivation for attacking a sensor network. In our definition, what we are trying to protect is *what the system sees*, and thus the ability to *predict how the user of the system might react*, as opposed to merely protecting *information about the environment*. We note that our methods can generalize to handling the latter case as well: the two definitions will essentially coincide if we constrain our sensor network application to return the most accurate information possible about the environment.

In the next two sections, we first define our network model and means for determining cost; then we discuss how we evaluate networks' vulnerability to eavesdropping — first for height-two aggregation trees, and then for trees of arbitrary depth.

3. SENSOR NETWORK MODEL

We begin by introducing our model of a sensor network, beginning by examining how computation is performed, and then quantifying the quality (accuracy) of the network and its cost. These factors, as well as the vulnerability of the network to eavesdropping (next section) will form the basis of assessing sensor networks.

3.1 Streams and Aggregation

Data from sensors is typically continuous and time-varying, as opposed to actually having discrete values; a formal stream model, similar to that of [1], is appropriate to capture this aspect of data.

DEFINITION 1. *(Sensor Stream) A Sensor Stream R is a possibly infinite sequence of elements, $\{\langle id, d, \tau, \rho\rangle_n\}_{n\geq 1}$, where $id \in \mathbb{Z}^+$ is a identifier for the sensor, d is a sensor data structure, τ is the timestamp and ρ is either $\emptyset$ or the location of the sensor.* □

We reason about two orthogonal types of aggregation over streams: *in-stream* aggregation, which occurs over a single stream, generally over a time window, and *multi-stream* aggregation, which occurs across the values of multiple streams, either at the same time or over a time window.

In-stream aggregation can be thought of as aggregation over all data from a single sensor within some time window. We can also define aggregation over streams of data from different sensors within the same time window. We refer to this form of aggregation as multi-stream aggregation.

3.2 Hierarchical Aggregation

For purposes of formal analysis, we abstract away specific details of sensing, communication and computation and view the network from a pure data collection and aggregation perspective. The hierarchical aggregation tree is a recursive structure in which, at each level of the tree, groups of child nodes send their values to a parent node that aggregates their values. The base station is the intermediate point at the highest level. Our model is consistent with most proposed aggregation algorithms, e.g. [16, 23, 11].

Finally, we assume that the values observed at each sensor are not identical, but can be characterized according to some probabilistic data distribution. Data from a sensor network will typically consist of a number of observed attributes; a *probability density function* (pdf) can be used to assign a probability for each possible assignment to the attributes. Such a model can be learned from data collected over time, using algorithms such as those in [17]. Learning a model involves maintaining certain parameters, e.g., the mean and the variance, and coping with noise, outliers, etc. A significant literature exists on learning models of streams, (e.g., [3, 5]).

Many sensor applications include multiple, dynamic attributes, and hence correlations and temporal aspects to the data distribution must also be considered. In [8], the authors used Markovian models to learn the time-varying effects of sensor readings. In their model, given the value of all attributes at time t, it is assumed that the value of the attributes at time $t+1$ are independent of those for any time earlier than t. This is generally sufficient to capture the dynamic nature of the sensor data. The same authors have extended their work in [7] to consider correlations between streams.

Our work assumes that such distributions are given (or can be reasonably approximated). Based on knowledge of the data distribution, we can provide specific probabilistic guarantees about sensing and eavesdropping. Additional information, such as the spatial distribution of sensors, is not assumed, although it can add to the precision of the metrics we present.

We illustrate an example aggregation tree in Figure 1, where nodes $s_0, \ldots, s_5$ are in a hierarchical group. Each of $s_1, \ldots, s_5$ perform aggregation of data in sub-groups and combine their own data with this before forwarding it to node s_0. Node s_0, in addition to recording its own sensor data, is also the final aggregator for all the data in the network.

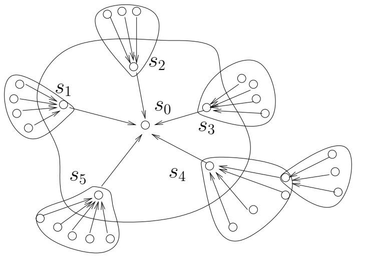

Figure 1: Sensor network model

We consider the presence of a powerful adversary who has the capability of listening to the messages in the sensor network, or of compromising sensor nodes in an undetectable way, with a certain probability. The higher a compromised node is in the aggregation tree, the more power the attacker has.

Notation. We denote the set of all sensor data streams within a group in the hierarchical network with the symbol S. Some subset, $S_C \subseteq S$ of these data values will be used to compute the stream aggregate σ (this quantity considers the possibility of dropped messages, filtering, sampling, etc.). The adversary can eavesdrop on some set of nodes $S_A \subseteq S$, which may overlap with but differ from S_C.

EXAMPLE 1. *Consider, the situation depicted in Figure 1, where the top-level group of an activity monitoring sensor network has nodes $s_0, \ldots, s_5$. Assume the sensors $s_1, \ldots s_5$ perform their local aggregation tasks and output their values to node s_0 once every 5 seconds. Also assume that the values from all data streams have the in-stream aggregation function σ_1 to be the mean of all the readings obtained at each node s_i over the past 4 sampling intervals. Let the multi-stream aggregation (σ_2) be applied every 20 seconds, as the mean of the readings from $s_0, \ldots, s_5$.*

If readings from s_0 are $\{4.82, 4.81, 4.82, 4.83\}$, then $\sigma_1(s_0) = 4.82$. Similarly, if $\sigma_1(s_1) = 4.93$, $\sigma(s_2) = 5.17$, $\sigma_1(s_3) = 4.92$, $\sigma_1(s_4) = 4.87$, $\sigma_1(s_5) = 5.04$ and we compute the mean over all streams, then $\sigma_2(S) = 4.96$. □

3.3 Quality of the Sample

Given a model of the distribution of data readings in the environment, there are several possible metrics for estimating the quality (accuracy) of the sample. We assume that the readings used to produce a single aggregate stream element occur within some time window $[T, T+\Delta]$. The length of the window, Δ, is application-specific, and it corresponds to the common notion of an *epoch* [16] during which computations are performed, but it allows readings to occur at any point within the window.

In statistics, goodness-of-fit is used to measure the distance between the data and the hypothesis. For example, if the underlying distribution is normal, then goodness-of-fit can be determined by using the standard χ^2 test. We adopt a statistic that works better for small samples and is simple to compute, the *Kolmogorov-Smirnov test* [12]. To compare a data sample consisting of N events whose cumulative distribution is $S_N(x)$ with a hypothesis function whose cumulative distribution is $\Phi(x)$, the value η is calculated as $\eta = \max_x |S_N(x) - \Phi(x)|$.The Cramer-Smirnov-Von-Mises test is often used to test that a one-dimensional data sample is compatible with being a random sampling from a given distribution: If the density function of the data is $f(x)$, then, the test measures the goodness-of-fit by the measure W^2, which is given by $\int_{-\infty}^{\infty} [S_N(x) - F(x)]^2 f(x) dx$. There are many alternative tests, depending on the distribution of data; for details we refer the reader to a standard textbook on statistics (e.g., [12]).

EXAMPLE 2. *If we assume that the data in Example 1 is distributed N(5,0.1) and use the χ^2 test as the goodness-of-fit measure, we have $\Delta = 20s$ and $\sum_{i=0}^{5} \frac{(\bar{s_i} - 4.96)}{0.1} = 0.911$, which implies that we have a sample close to the actual model.* □

3.4 Cost of Sensing

We can estimate the cost of producing a single output element in the sensor network by considering the cost of acquiring and communicating the sensor readings. Let the time window be $\mathcal{T} = [T, T+\Delta]$, the cost of acquiring a reading at sensor node s be $c_a(s)$, and the cost of transmitting a message from sensor s to the aggregating point s_0 be $c_t(s)$. Then the cost of acquiring the data to be aggregated is $\mathcal{C}_a(\mathcal{T}, S) = \sum_{s \in S} c_a(s)$. For each intermediate node in the aggregation tree, the cost of transmitting sensor data is $\mathcal{C}_t(\mathcal{T}, S) = \sum_{s \in S} c_t(s)$, where $c_t(s_0) = 0$. (This is because there is no transmission involved from s_0 to itself). Let the reception cost for one reading at s_0 be c_r. Then, the total cost of reception $\mathcal{C}_r(\mathcal{T}, S \backslash S_0) = |S \backslash S_0| \cdot c_r$ where S_0 is the set of readings obtained at s_0. Thus, the total cost for acquiring and aggregating the data is $\mathcal{C}(\mathcal{T}, S) = \mathcal{C}_a(\mathcal{T}, S) + \mathcal{C}_t(\mathcal{T}, S) + \mathcal{C}_r(\mathcal{T}, S \backslash S_0)$ for any set S of nodes that share a single aggregation point s_0.

EXAMPLE 3. *Let us assume that the cost of sensing for attribute is $0.015J$ and transmitting and receiving data takes $0.025J$ of energy for all the sensors in Example 1. In one epoch, the sensors transmit $\frac{20}{5} = 4$ packets. Hence, $\mathcal{C}(S) = 5 \times 4 \times (0.025 + 0.015) + 4 \times 0.015J + 5 \times 4 \times 0.025J = 1.36J$.* □

4. MODELING EAVESDROPPING

We now consider the case of an adversary who has access to some of the sensor readings (either through eavesdropping or compromise), and who is trying to determine the aggregate value output by the sensor network.[1] We consider the confidentiality of the network, in terms of whether the adversary can estimate the output

[1] As described in Section 2, this definition is motivated by the fact that the eavesdropper is most likely to be interested in predicting the behavior of the person or application monitoring the sensor data.

value within some small tolerance δ. We compute the *eavesdropping vulnerability* based on several important parameters. First, there is the probability that a compromised set of sensor nodes, S_A, greatly resembles the set of nodes that our application is sampling, S_C. This probability is a function of the size of S_C, the specific aggregate function σ, and the data distribution of the sensors S. For example, if all sensors produce the same reading, then the adversary can compromise the system from a single reading. We formalize the probability based on these parameters.

DEFINITION 2. *(Eavesdropping Vulnerability) The eavesdropping vulnerability (γ) relative to a set of compromised nodes is defined as $\gamma(\sigma, S, S_A, S_C, \delta) = p\left(|\sigma(S_C) - \sigma(S_A)| \leq \delta\right)$, where σ is the aggregating function and δ the adversary's error tolerance.* □

Although we have considered a single aggregate computation here, the eavesdropping vulnerability can be generalized to support multiple aggregate computations over different attributes: the expected value of γ can be obtained by conditioning on different parameters.

We can compute the *expected* eavesdropping vulnerability, in which the specific S_A is unknown, as $\bar{\gamma} = \sum_s p(S_A = s) \cdot \mathcal{I}(|\sigma(S_C) - \sigma(s)| \leq \delta)$, where $\mathcal{I}$ is an indicator function that evaluates to 1 if the condition is true and 0 otherwise.

This relies on knowledge of the underlying sensor value distribution of S, and the specific aggregation function, σ. We now show the derivation of γ values for the most common sensor aggregation functions (min, max, sum, avg and $median$) over single attributes with discrete distributions:

- *Min/Max:* $\mathcal{I}(|\min(S_C) - \min(S_A)| \leq \delta) = 1$ if $\min(S_A)$ lies between $[\min(S_C) - \delta, \min(S_C) + \delta]$. If f is the probability density function (pdf) and Φ is the cumulative density function (cdf) of the distribution of S, then, for any j, $f(j)$ is the probability of obtaining a j and $(1 - \Phi(j))$ is the probability that a reading is greater than j. Thus in a sample of size i, j will be the minimum with probability $f(j)\,(1 - \Phi(j))^{i-1}$. Therefore:

$$\bar{\gamma} = \sum_{i=1}^{|S|} p(|S_A| = i) \sum_{j=\lceil \min(S_C)-\delta \rceil}^{\lfloor \min(S_C)+\delta \rfloor} f(j) \cdot (1 - \Phi(j))^{i-1} \tag{1}$$

  Using a similar argument for *Max*, we get:

$$\bar{\gamma} = \sum_{i=1}^{|S|} p(|S_A| = i) \sum_{j=\lceil \min(S_C)-\delta \rceil}^{\lfloor \min(S_C)+\delta \rfloor} f(j) \cdot \Phi(j)^{i-1} \tag{2}$$

- *Sum:* $\mathcal{I}(|sum(S_C) - sum(S_A)| \leq \delta) = 1$ if $sum(S_A)$ lies between $[sum(S_C) - \delta), sum(S_C) + \delta)]$. If $f_{|S_A|}$ is the pdf of the sum of variables and $\Phi_{|S_A|}$ is the cdf of the sum of variables, we get:

$$\bar{\gamma} = \sum_{i=1}^{|S|} p(|S_A| = i) \cdot \left(\Phi_{|S_A|}(u) - \Phi_{|S_A|}(l)\right) \tag{3}$$

  where $u = (sum(S_C) + \delta)$ and $l = (sum(S_C) - \delta)$.

- *Avg:* $\mathcal{I}(|avg(S_C) - avg(S_A)| \leq \delta) = 1$ if $sum(S_A)$ lies between $[|S_A|(avg(S_C) - \delta), |S_A|(avg(S_C) + \delta)]$. If $f_{|S_A|}$ is the pdf of the sum of variables and $\Phi_{|S_A|}$ is the cdf of the sum of variables, then with a similar argument as before, we get:

$$\bar{\gamma} = \sum_{i=1}^{|S|} p(|S_A| = i) \cdot \left(\Phi_{|S_A|}(u) - \Phi_{|S_A|}(l)\right) \tag{4}$$

  where $u = i(avg(S_C) + \delta)$ and $l = i(avg(S_C) - \delta)$.

- *Median:* $\mathcal{I}(|med(S_C) - med(S_A)| \leq \delta) = 1$ if $med(S_A)$ lies in $[med(S_C) - \delta, med(S_C) + \delta]$. If f be the probability density function (pdf),and Φ is the cumulative density function (cdf) of distribution of S, then, for any j, $f(j)$ is the probability of obtaining a j, $\Phi(j)$ is the probability that a reading is less than j, and $(1 - \Phi(j))$ is the probability that a reading is greater than j. Thus in a sample of size i, j will be the median with probability, $p(j) = \binom{i}{\lfloor \frac{i}{2} \rfloor} \cdot f(j) \cdot \Phi(j)^{\lfloor \frac{i}{2} \rfloor} \cdot (1 - \Phi(j))^{i - \lfloor \frac{i}{2} \rfloor - 1}$. Therefore:

$$\bar{\gamma} = \sum_{i=1}^{|S|} p(|S_A| = i) \sum_{j=\lceil \min(S_C)-\delta \rceil}^{\lfloor \min(S_C)+\delta \rfloor} p(j) \tag{5}$$

□

EXAMPLE 4. *To evaluate the expected value of γ for the application in Example 1, let us assume that the probability of the adversary eavesdropping on a single node is 0.2 and the data is distributed as $N(5, 0.1)$. Also, let the tolerance $\delta = 0.1$. Noting that we have $\sigma_2(S) = 4.96$, we can use Equation (4) to evaluate the expected probability. We get $\bar{\gamma} = \sum_{i=1}^{5} p^i \cdot (\Phi(5.06) - \Phi(4.86))$ [2], which on evaluation yields $\bar{\gamma} = 0.2499$. This agrees with our intuition that if the adversary is able to compromise one node, then she is far from being able to estimate the aggregate of the network consisting of 5 nodes.* □

4.1 Hierarchical Aggregation

Thus far, we have only considered aggregation within a group with a single aggregation point. We now generalize to eavesdropping over hierarchical groups: the goal is to consider how close the adversary gets to an aggregate value higher in the tree when she eavesdrops on data in the lower levels comprising that group. (If we assume that the adversary eavesdrops only at one level, then this problem is identical to the one considered above.) The higher the adversary listens, the closer she gets to aggregate of the whole network.

An example scenario is depicted in Figure 1, where we assume that the adversary has eavesdropped on groups with nodes $s_1, \ldots, s_5$ as the nodes responsible for aggregation. Now, we want to know how close she gets to the aggregate at s_0.

The probability of adversary learning the result of aggregation at a level l is called the *eavesdropping vulnerability over a hierarchy* and is denoted by γ_l, where l indicates the hierarchical level from which the adversary listens with the goal of compromising the overall system. As with γ, γ_l will be a function of $S^l, S^l_A, S^l_C, \sigma$ and δ. We consider the effect of a lower-level compromise on a higher-level node to be a "partial compromise" of the higher node, i.e., we define $S^l_A = \bigcup_i \sigma(S^{l-1}_{A_i}), l > 1$. Note that the adversary's set at level l is the union of sets $\sigma(S^{l-1}_{A_i})$, which accounts for the fact that the sensor values at level l will be aggregates of values at level $l - 1$.

[2] These values can be found by converting it into standard normal form for which Φ is well tabulated.

DEFINITION 3. *(Eavesdropping Vulnerability over a Hierarchy) The eavesdropping vulnerability (γ_l) for the adversary over a hierarchy is defined as $\gamma_l(\sigma, S^l, S^l_A, S^l_C, \delta) = p\left(|\sigma(S^l_C) - \sigma(S^l_A)| \leq \delta\right)$, where σ is the aggregating function and δ is the error in estimate, and $S^l_A = \bigcup_i \sigma(S^{l-1}_{A_i}), l \geq 1$.* □

Note that with this definition, $\gamma = \gamma_0$. We can compute γ_l by conditioning on various parameters. For example, knowing σ, S^l, S^l_C and δ, we can compute:

$$\overline{\gamma_l} = \sum_{S^{l-1}_{A_1},\ldots,S^{l-1}_{A_n}} p(S^{l-1}_{A_1}, \ldots, S^{l-1}_{A_n}) \cdot \mathcal{I}(d \leq \delta) \tag{6}$$

where $d = |\sigma(S^l_C) - \sigma(S^l_A)|$.

Computing γ_l, in general, involves knowing how much the data from different groups are related. If the data from different groups at level $l-1$ are correlated, then computing γ_l can be quite difficult. Correlations between groups are also undesirable because they can help the adversary can make a good estimate by eavesdropping on only a few groups.

Although the exact computation of γ_l is generally difficult, an approximate answer by making some simplifying assumptions, such as simultaneous eavesdropping in all the groups. The example below illustrates this idea.

EXAMPLE 5. *Consider the scenario in Example 1. Let us assume that each of the nodes $s_0, \ldots, s_5$ are themselves aggregating data in their groups and that the distribution in each group is as follows: $s_1 : N(4.9, 1)$, $s_2 : N(4.8, 1)$, $s_3 : N(4.8, 1)$, $s_4 : N(5, 1)$, $s_5 : N(5.2, 1)$, and the data from node s_0 is distributed $N(5, 1)$. If the data at this level is being averaged, the resulting average will be normally distributed with mean $\frac{5+4.9+4.8+4.8+5+5.2}{6}$ and a standard deviation $\frac{1+1+1+1+1+1}{36}$, which has distribution $N(4.95, 0.16)$. Now, if the probability of eavesdropping simultaneously in every group is 0.5, the eavesdropping vulnerability for $\delta = 0.1$ is $\sum_{i=1}^{5} (0.5)^i \cdot (\Phi(5.06) - \Phi(4.86)) = 0.4599$.* □

4.2 Performance Ratio

The eavesdropping vulnerability γ or γ_l gives us the probability that an adversary can obtain a good estimate of the actual aggregate. Obviously, we would like to design sensor networks that minimize this probability; however, to do this, we will generally have to incur additional overhead.

If we use *benefit* to mean how close an estimate is to the target (in the case of our application, this is the "real" aggregate; in the case of the adversary, this is our network's aggregate), we can define a *performance ratio* to compare different sensor network schemes. We define the performance ratio of the adversary relative to a set of compromised nodes, ρ_A, as: $\rho_A(\sigma, S, S_A, S_C, \delta, \mathcal{C}) = \frac{\gamma(\sigma, S, S_A, S_C, \delta)}{\mathcal{C}_r(S_A)}$. The increase in cost incurred to reduce γ can be measured by $\frac{\mathcal{C}(S)}{\mathcal{C}'(S)}$. Here, $\mathcal{C}'$ is the cost model for any eavesdropping-tolerant data protocol and $\mathcal{C}$ is the cost model for the standard streaming model, as defined earlier. We can now define the *performance ratio of a sensor network, ρ*, as:

$$\rho(\sigma, S, S_A, S_C, \delta, \mathcal{C}, \mathcal{C}') = \frac{1}{\rho_A(\sigma, S, S_A, S_C, \delta, \mathcal{C})} \cdot \frac{\mathcal{C}(S)}{\mathcal{C}'(S)} \tag{7}$$

We can calculate the expected value of ρ by conditioning on various parameters. Ideally, we would like to design our data protocol to maximize ρ as much as possible.

EXAMPLE 6. *Consider the application in Example 1 with the cost as computed in Example 3. We assume that the probability of the adversary eavesdropping on a node is 0.2, yielding a cost of $0.025 * 4J = 0.1J$. $\sum_{i=1}^{5} \frac{(\Phi(5.06)-\Phi(4.86)) \cdot p^i}{0.1i} = 1.799$. $\bar{\rho}$ can now be computed as, $\frac{1}{1.799} \cdot \frac{1.36}{1.36} = 0.5558$. Intuitively, higher cost for the adversary increases the ratio ρ. If we make it harder for the adversary to eavesdrop, say reducing the probability of eavesdropping on a single node to 0.1, then we will have, $\bar{\rho} = 1.2248$. Techniques for increasing performance ratio are discussed in the next section.* □

To increase the quality of the sample, we need more observations (S_C), which, however, increases both cost and (if the distribution of values remains the same) γ. Hence, we can identify a trade-off between quality, cost, and having a eavesdropping vulnerability.

5. COUNTERMEASURES AGAINST EAVESDROPPING

Given our understanding of the factors that affect eavesdropping potential, we now present some general techniques to thwart adversaries. We distinguish between traditional, cryptographic techniques and non-cryptographic schemes.

5.1 Cryptographic techniques

Encryption and authentication using cryptographic techniques makes a system significantly more secure against eavesdropping and other attacks. Encryption can be used to keep data secure from the adversary, and authentication can be used to safeguard against spurious data. In essence, these techniques attempt to ensure system-level confidentiality by protecting all links. For the sensor network environment, symmetric key techniques are most commonly used, but it is unclear how to manage keys and how to justify the overhead of encryption. Among the many prior works on cryptographic techniques for privacy in wireless sensor networks, [18] and [15] describe methods to achieve authenticity and confidentiality.

However, many approaches (e.g., [19, 9]) assume a pre-key distribution which impedes network creation and makes dynamic membership difficult. In [4], Chan and Perrig advocate that end-to-end encryption is not possible for sensor networks and foresee new methods as the solution. Moreover, encryption may not help if the nodes themselves can be compromised. Taking our cue from these points, we briefly suggest several alternatives below.

5.2 Non-cryptographic techniques

Non-cryptographic techniques make it harder to eavesdrop by reducing the chance that an adversary's sensor data sample S_A matches the system's sample S_C.

Data Filtering or Compensation. One technique is to deliberately send spurious data (or data with spurious offsets) from the sensors, and to filter the noise at the aggregating point. After filtering, the resulting data set will comprise legitimate information about the underlying network. The adversary, who is not aware of this shared information, will see data that follows a different distribution.

One such idea, which we are investigating extensively, is termed *confusion* [6]. Under such a scheme, whenever the sensor wishes to transmit a message, it appends the shared secret (token) to the message. A set of confusion-generating nodes then could inject spurious data, which is indistinguishable to a third party, into the network. Such confusion messages could be generated either by third party nodes or be a subset of sensors themselves. At the receiving end, the secret can be used to separate the legitimate message from the noise. Yet while the aggregate node can filter out superfluous messages from confusers, an eavesdropper with incomplete knowledge cannot make such distinctions. Since the eavesdropper is not aware of which tokens belong to the sensors and which belong to the confusers, she cannot identify the legitimate messages. Thus, if she ends up accepting the "noise," she will end up with a different distribution of the data in the network.

As with encryption techniques, a confusion-based technique assumes a shared secret unique to a sensor, but it may may require less computational power per sensor node, it is tolerant to the compromise of a few nodes, and it is resistant to active eavesdropping. The savings on per-device power in the confusion-based approach comes from the fact that there is no need for the expensive exponentiation operations involved in encryption. Confusion does require more message transmissions, but these can be amortized by adding greater numbers of devices.

EXAMPLE 7. *Consider the application in Example 1. Suppose the sensors double their transmission rate by injecting a spurious value for every legitimate one. Assume that the legitimate data is distributed within the range* $N(5, 0.1)$*, while the spurious data ensures the adversary's sample will be uniformly distributed in* $[4, 7]$*. Given the model of Example 6, the cost is* $\mathcal{C}(S) = 8 \times 5 \times (0.025 + 0.015) + 4 \times 0.015 + 8 \times 5 \times 0.025J = 2.66J$*.* $\sum_{i=1}^{5} \frac{(\Phi(5.06)-\Phi(4.86))\cdot p^i}{0.1i} = 0.1492$*.* $\bar{\rho}$ *can now be computed as,* $\frac{1}{0.1492} \cdot \frac{1.36}{2.66} = 3.4267$*. Clearly, this technique greatly reduces the vulnerability of the network, when compared to the baseline model's* $\bar{\rho} = 0.5558$*.*

Data cloaking [10] has been proposed as another approach to achieving privacy in sensor networks. Cloaking of data involves perturbing the data by a predefined offset. This has been used to achieve anonymity within a network. A similar idea can also be used to counter eavesdropping: 1) First, nodes are partitioned into disjoint subsets. 2) Then, based on a shared secret, each node within a partition is assigned an offset. This offset is added to the actual sensor reading before transmission. Ideally, this offset should be unique to a partition. 3) At the point of aggregation, the appropriate offset is subtracted from the reading before aggregation.

Although this scheme requires maintaining a node-to-offset mapping at the aggregating point, it can easily be obviated by having all the nodes within a partition transmit within a time slot. With such a routing protocol, only the mapping of different time slots to the offset would have to be stored and this information is modest compared the original mapping.

The adversary, who has no information about the offset, will be readily misled by the transmitted information. Even if she manages to compromise a few nodes and learn the offset information, the damage is limited to members of the partition with the compromised nodes.

EXAMPLE 8. *Consider the scenario in Example 1. Let us assume that the nodes* $s_1, \ldots, s_5$ *are themselves aggregation point of their groups and their data is distributed as* $N(5, 1)$*. Also, let the data at node* s_0 *be also distributed* $N(5, 1)$*. If average is the aggregation function used, it will be normally distributed with mean* $\frac{5 \times 6}{6} = 5$ *and standard deviation* $\frac{1 \times 6}{36} = 0.16$*. If we assume that the probability of eavesdropping on a single message is* 0.5*, the eavesdropping vulnerability is* $\sum_{i=1}^{5} (0.5)^i \cdot (\Phi(5.06) - \Phi(4.86)) = 0.9843 \times 0.4553 = 0.4482$*.*

Now, if we assume that each sensor $i, i \in \{0, \ldots, 5\}$ *adds an offset* $0.1i$*, which is subtracted out at* s_0*, then the average will be normally distributed with mean* $\frac{5+5.1+5.2+5.3+5.4+5.5}{6} = 5.25$ *and standard deviation* $\frac{1 \times 6}{36} = 0.16$*. In this case, the eavesdropping vulnerability will be* $\sum_{i=1}^{5} (0.5)^i \cdot (\Phi(5.06) - \Phi(4.86)) = 0.9843 \times 0.1101 = 0.1083$*. which is a clear reduction in eavesdropping vulnerability from* 0.4482 *without using the offsets.* □

Attribute-value Correlation. Yet another possibility is to use correlations between different attributes. If the application at hand is temperature monitoring and a sensor's temperature and voltage are correlated, then, for instance, the sensors might transmit voltages in certain cases, and temperatures the remainder of the time. If we assume that the adversary does not have the correlation model, then such data will be useless to her. Constructing correlations between attributes has been previously studied (e.g., [7]), with the objective of reducing the cost for the network. Here we use it as shared information. Importantly, it takes a considerable amount of time, energy, and node samples to learn this correlation model, meaning that an attacker would need to devote significant resources to compromising a large portion of the system.

EXAMPLE 9. *Again consider Example 7, with the modification that with probability* 0.5*, the sensors send voltage readings. Further, they also output as many spurious messages as temperature readings, in order to ensure that the adversary's distribution is uniformly distributed in* $[4, 7]$*. In this case,* $\mathcal{C}(S) = (\frac{1}{2} \times 8 + \frac{1}{2} \times 4) \times 5 \times (0.025 + 0.015) + 4 \times 0.015 + \frac{1}{2} \times 8 + \frac{1}{2} \times 4) \times 5 \times (0.025) = 1.76J$*,* $\bar{\rho}$ *can now be computed as,* $\frac{1}{0.1492} \cdot \frac{1.36}{1.76} = 5.1791$*. which is better than strictly using the filtering/compensation approach.*

6. RELATED WORK

Prior works on sensor security [22, 14] present attack models, but our focus and attack taxonomy are a more general classification based on the *goals* of the adversary, and our focus is on the security of the overall system even when individual nodes are compromised.

There is also a significant literature on quantifying security in a context-specific way. [13] presents a quantitative model of the security intrusion process based on attacker behavior: their model is based on empirical data collected from intrusion experiments. [20] quantifies security strength and risk using economic criteria. It should be noted that though these are general methods, their applicability to sensor networks is uncertain. Our approach, in contrast, is based on data models for different applications of sensor networks.

The idea of developing a probabilistic model for data aggregation in sensor networks was introduced in [8]. We can use the same techniques to learn a model from the data. However, our focus is on using the model to understand the security vulnerabilities of a sensor network, as opposed to minimizing power usage in computing aggregates. This slightly resembles the resilient techniques for data aggregation of [21], although we focus on quantitatively ascertaining robustness in the presence of an adversary.

7. CONCLUSIONS AND FUTURE WORK

We have presented an attacker taxonomy for sensor networks which has three main classes of attackers: *eavesdropping*, *disruption*, and *hijacking*. So far as we know, our work is the first to focus on *quantifying* system-level eavesdropping vulnerability. We first study a single-level aggregation tree (γ) and then a hierarchical network (γ_l), developing a probabilistic scheme for assessing their eavesdropping vulnerability. We then consider trading off power consumption versus security and data quality/accuracy. Finally, we propose a series of solutions using cryptographic techniques, data filtering, and attribute correlation.

This paper represents an initial step in a much broader plan. First, we are extending our model to the disruption and hijacking models. We are also developing a comprehensive characterization of common sensor network protocols and aggregation functions with respect to their robustness. We ultimately hope to consider a range of other issues, such as unreliable networks, temporary outages, and correlations between the values at different sensors.

8. REFERENCES

[1] A. Arasu, S. Babu, and J. Widom. The CQL continuous query language: Semantic foundations and query execution. *Technical Report 2003-67, Stanford University*, 2003.

[2] S. Avancha, J. L. Undercoffer, A. Joshi, and J. Pinkston. Secure sensor networks for perimeter protection. *Computer Networks*, 43(4):421–435, November 2003.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, New York, NY, USA, 2002. ACM Press.

[4] H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer Magazine*, pages 103–105, 2003 2003.

[5] F. Chu, Y. Wang, and C. Zaniolo. An adaptive learning approach for noisy data streams. In *ICDM*, pages 351–354, 2004.

[6] E. Cronin, M. Sherr, and M. Blaze. On the reliability of internet eavesdropping, February 2005. Personal Communication.

[7] A. Deshpande, C. Guestrin, S. Madden, and W. Hong. Exploiting correlated attributes in acqusitional query processing. In *ICDE 2005*, 2005.

[8] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *30th VLDB Conference*, 2004.

[9] W. Du, J. Deng, Y. S. Han, S. Chen, and P. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *Proceedings of The 23rd Conference of the IEEE Communications Society*, 2004.

[10] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor networks. In *Proceedings of HotOS'03: 9th Workshop on Hot Topics in Operating Systems*, pages 163–168. USENIX, May 2003.

[11] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Towards sophisticated sensing with queries. In *2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, March 2003.

[12] I.Miller and J.E.Freund. *Probability and Statistics for Engineers, 2nd edition*. Prentice Hall,Inc, Englewood Cliffs, NJ., 1977.

[13] E. Jonsson and T. Olovsson. A quantitative model of the security intrusion process based on attacker behavior. *IEEE Trans. Softw. Eng.*, 23(4):235–245, 1997.

[14] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *IEEE Int'l Workshop on Sensor Network Protocols and Applications*, pages 113–127, May 2003.

[15] Y. W. Law, S. Etalle, and P. H. Hartel. Assessing Security-Critical Energy-Efficient sensor networks. In *Conf. on Security and Privacy in the Age of Uncertainty (SEC)*, pages 459–463, May 2003.

[16] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Design of an acquisitional query processor for sensor networks. In *SIGMOD 2003*, pages 491–502, 2003.

[17] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[18] A. Perrig, R. Szewczyk, V. Wen, D. E.Culler, and J. D. Tygar. SPINS: security protocols for sensor netowrks. In *Mobile Computing and Networking*, pages 189–199, 2001.

[19] B. Przydatek, D. Song, and A. Perrig. SIA: secure information aggregation in sensor networks. In *SenSys '03*, pages 255–265, 2003.

[20] S. E. Schechter. Computer security strength & risk: A quantitative approach. *Harvard University Doctoral Dissertation*, 2004.

[21] D. Wagner. Resilient aggregation in sensor networks. In *SASN: Proc. Workshop on security of ad hoc and sensor networks*, pages 78–87, 2004.

[22] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[23] Y. Yao and J. Gehrke. Query processing for sensor networks. In *CIDR 2003*, 2003.