2012 IEEE 28th International Conference on Data Engineering

A Game-Theoretic Approach for High-Assurance of Data Trustworthiness in Sensor Networks

Hyo-Sang Lim [#*1], Gabriel Ghinita [†2], Elisa Bertino [#3], Murat Kantarcioglu [‡4]

[#] *Dept. of Computer Science, Purdue University*
[†] *Dept. of Computer Science, University of Massachusetts, Boston*
[*] *Computer and Telecommunications Engineering Division, Yonsei University*
[‡] *Dept. of Computer Science, University of Texas at Dallas*
[1] hyosang@yonsei.ac.kr, [2] gabriel.ghinita@umb.edu
[3] bertino@purdue.edu, [4] muratk@utdallas.edu

***Abstract*—Sensor networks are being increasingly deployed in many application domains ranging from environment monitoring to supervising critical infrastructure systems (e.g., the power grid). Due to their ability to continuously collect large amounts of data, sensor networks represent a key component in decision-making, enabling timely situation assessment and response. However, sensors deployed in hostile environments may be subject to attacks by adversaries who intend to inject false data into the system. In this context, *data trustworthiness* is an important concern, as false readings may result in wrong decisions with serious consequences (e.g., large-scale power outages). To defend against this threat, it is important to establish trust levels for sensor nodes and adjust node trustworthiness scores to account for malicious interferences.**

In this paper, we develop a game-theoretic defense strategy to protect sensor nodes from attacks and to guarantee a high level of trustworthiness for sensed data. We use a discrete time model, and we consider that there is a limited attack budget that bounds the capability of the attacker in each round. The defense strategy objective is to ensure that sufficient sensor nodes are protected in each round such that the discrepancy between the value accepted and the truthful sensed value is below a certain threshold. We model the attack-defense interaction as a Stackelberg game, and we derive the Nash equilibrium condition that is sufficient to ensure that the sensed data are truthful within a nominal error bound. We implement a prototype of the proposed strategy and we show through extensive experiments that our solution provides an effective and efficient way of protecting sensor networks from attacks.

I. Introduction

Sensor networks are being increasingly deployed in many application domains ranging from monitoring the environment (e.g., measuring pollution levels or detecting earthquake activity) to controlling automated systems such as manufacturing facilities or power plants. Due to their ability to continuously collect large amounts of data and stream them to applications, sensor networks represent a key component in decision-making infrastructures, enabling timely situation assessment and response. On the other hand, since sensed data may be used in critical processes, the requirement for correct data can not be overemphasized.

[0]The majority of work by G. Ghinita was done while he was at Purdue University.

In practice, incorrect or false data may be injected in the network as a result of device malfunctioning or malicious interference from attackers. Therefore, an essential task is to devise mechanisms to measure the amount of *data trustworthiness* for sensor readings and to filter information such that only highly-trusted data are delivered to the application. To address sensor malfunctioning, well-understood fault-tolerance principles may be applied to correct or filter out erroneous data. On the other hand, establishing data trustworthiness in the presence of malicious adversaries is a more challenging task. In this work we focus on high assurance of data trustworthiness in adversarial scenarios.

To underscore the importance of assessing data trustworthiness, consider the following two motivating applications:

1) *The Power Grid.* Electric power transmission networks represent a critical infrastructure component. To ensure that electric energy is delivered to end users with the proper parameters, Supervisory Control And Data Acquisition (SCADA) systems collect real-time information from sensors that monitor voltage, current intensity, etc. To maintain operating parameters within their nominal range, the system makes decisions based on the sensed data, such as increasing or decreasing the output power at the generation plant. Typically, data are collected from remote sensors [8] which may be subject to damage due to meteorological phenomena (e.g., storms). Furthermore, malicious adversaries (e.g., terrorist elements) may take over a subset of the monitoring sensors and inject false data with the purpose of overloading and crashing the power grid.
2) *Battlefield Monitoring Systems.* Sensors deployed on enemy territory can gather the locations of enemy soldiers, vehicles, etc., and report them back to the command center. If some of the sensors are captured by the enemy, they can be used to inject fake information and interfere with the military strategy of the sensor deploying party. Assessing the trustworthiness of reported locations can help filter out wrong data such that mission critical applications access only highly trusted sensor readings,

1084-4627/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDE.2012.78

therefore ensuring that correct offensive decisions are made.

In this work, we will focus on the former scenario, since the power grid is a critical infrastructure with direct impact on the day-to-day life. For instance, a catastrophic failure which occurred in 2003 resulted in a blackout [1] over the Midwest and Northeast of the United States as well as Ontario, Canada. A population of cca 50 million people were affected. The resulting financial losses are estimated to have ranged between $4 and $10 billion. Investigations showed that one of the reasons that led to the failure was bad telemetry data, which in turn rendered inoperative the power flow monitoring tool that controlled the electric network. This event shows the necessity of ensuring the secure and reliable collection of data in power grids. In a power grid, the state of the system is typically estimated based on the readings of meters placed at well-defined points in the grid. If an attacker compromises some meters, the resulting malicious data readings may trigger critical errors in the system state estimation, leading to system breakdown. On the other hand, if mechanisms are in place that can effectively assess sensed data trustworthiness, correct decisions can be made by ruling out the values detected to originate at untrustworthy sensors.

Current approaches to filtering out erroneous data in power grids rely on statistical analysis. Assume that a number of meters are placed along a power line, and that each meter reports the sensed value for the electrical current. According to Kirchhoff's circuit laws [21], the current should have the same value along a certain circuit segment. A significant body of research [5], [9] employs techniques for noise filtering, e.g., statistical estimators, that aim at eliminating the contribution of outlier sensor values. This is a good approach to deal with device malfunctions, but in the case of active attackers statistical methods may fail to protect against wrong readings. Consider the example in Fig. 1, where eight sensor nodes read the value of the electrical current. Assume that a simple statistical test known as 3σ is used, where all values that are more than three times the standard deviation away from the average are discarded. At round 1, all nodes are reliable, and the value accepted by the system (the average among all readings) is close to the actual value (small errors may occur due to device imperfections).

At round 2, an adversary compromises three nodes, and alters the readings of these values such that the 3σ interval is skewed towards lower values. Since three distinct nodes report a lower value, the statistical test will conclude that the rightmost node must be in error, since its value is outside the confidence interval. Therefore, its value is discarded, and the node is marked as less trustworthy for the next round. In the third round, the adversary shifts again its reported values, and manages to determine the system to declare the second rightmost node untrustworthy as well. This way, through careful selection of reported values, an attacker is able to circumvent the statistical test error detection technique. More importantly, the attacker manages to shift the accepted value far away from the actual value. If the difference exceeds

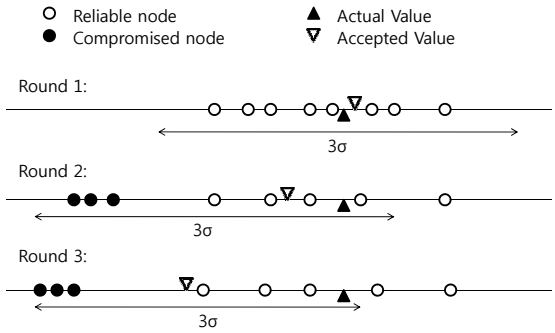


Fig. 1. Statistical filters do not detect malicious data injection.

a certain threshold (which is application dependent), then a catastrophic failure may occur.

To address the shortcoming of existing techniques, we focus on a defense strategy based on a game theory model which allows the system to determine how many, and which nodes need to be protected in order to ensure that the accepted value is never far away from the actual one. We consider two distinct protection models: first, we consider the case when once a node is protected, it is immune to attacks. Next, we relax this condition and consider that attacks may still have an impact on protected nodes, but only a limited impact compared to the case of unprotected nodes. Our defense strategy accounts for adversaries that continuously adapt their attack to obtain better results. For example, if the system deploys additional security controls to protect some parts of the sensor network, the attacker may change its strategy and attack unprotected sensors. We see such adaptive behavior in various real life scenarios. For example, increasing security measures at airports may cause terrorists to choose other targets (e.g., railway stations) that are not as heavily protected. Another important aspect that we consider is the tradeoff between the increased protection and the cost incurred to achieve that protection level.

Our game-theoretical defense formulation relies on the Stackelberg competition [10], a model according to which a game is represented by a sequence of actions performed by two entities: a leader and a follower. The defender (leader) protects the sensor network to a certain amount, whereas the attacker (follower) tries to take over sensors and inject malicious data into the network. This multiple-round interaction is repeated throughout the life-time of the sensor network. The reliability of the system can be guaranteed within this model, as long as there exists an upper bound on the attack budget available to the adversary, a reasonable assumption in practice.

The actions of the defender and attacker (i.e., the defense and attack strategies) are formulated with respect to *trustworthiness scores* of sensor nodes. Specifically, nodes that are detected to be compromised or report unusual values are assigned low trust scores, and these scores evolve over time to account for the reliability of the data sensed by each node. Based on trust scores, the defender decides which nodes to protect such that it is guaranteed that the attacker cannot possibly alter the accepted value beyond a certain threshold

within the given attack budget.

In summary, our contributions are:

- We formulate the problem of high assurance of data trustworthiness as a Stackelberg competition, under the assumption that the adversary has a limited attack budget. To the best of our knowledge, this is the first work to address data trustworthiness assurance using game theoretic models.
- We provide two alternate utility and cost measures for the defender-attacker game, which account for distinct models of sensor node protection and attacker budget.
- We develop a prototype defense system and show through experiments that our solution maintains high levels of data trustworthiness even against adaptive adversaries that continuously change the set of attacked nodes. We also show that the defense strategy computation has low overhead.

The rest of the paper is organized as follows. Section II introduces basic concepts used in our work. Section III outlines the game-theoretic model for the defense and attack interaction in sensor networks. Section IV provides solutions to find defense strategies, whereas Section V reports experimental results. We discuss related work in Section VI and conclude the paper in Section VII.

II. Preliminaries

In this section, we give an overview of the main concepts relevant to our work.

A. Sensor Networks

A sensor network is represented by an arbitrary graph of sensor nodes that gather process data and a server (or sink) node that receives data items and performs data analysis and/or decision-making. The edges in the graph represent communication links used in forwarding data from sensing nodes to the sink. We assume that a particular sensor node can influence (i.e., either generate or modify) only the data items collected by that node, and cannot modify the data forwarded from other sensor nodes. This can be achieved using secure sensor communication with cryptographic signatures as in TinyEcc [13].

For simplicity, we also assume that all the sensor nodes in the network collect data for a single event or system parameter. For example, as discussed in Section I, all sensors may measure the intensity of the electric current through a certa power distribution network segment. Note that, this sing event assumption does not limit the scope of our approac If there are multiple events or parameters monitored by tl network, then sensors can be partitioned according to tl targeted event, and each partition is handled separately usi the proposed defense strategy.

B. Data Trustworthiness Scores

The trust score represents a quantitative measure of trus worthiness ranging from 0 to 1, where 0 signifies no trust a particular data item, and 1 signifies complete trust. In o setting, we assess trust scores for both data items and sensor nodes. In fact, there exists an inter-dependence between the trust scores of data items and nodes: a node that consistently generates highly-trusted data will be assigned a high trust score. Conversely, data generated by trustworthy nodes are assigned high trust scores. Trust scores can be relevant both in terms of absolute values, as well as in terms of the ordering (i.e., ranking) they provide for a set of data items/nodes. For example, even though the meaning of absolute score values may vary depending on the application or parameter settings, it is possible to determine based on scores what is the most trustworthy data item in a data set. We adopt the trustworthiness assessment mechanism proposed in [12]. According to this trust model, given a set of sensor readings v_i each with trust score s_i, the value accepted by the sink is the weighted mean

$$v = \frac{\sum_i v_i \cdot s_i}{\sum_i s_i} \tag{1}$$

Initially, at the beginning of the sensing process, the trust score of a data item is inherited from the trust score of the sensor node which generated the data. At subsequent rounds, trust scores are adjusted by the server (sink) according to value similarities among sensing nodes. Here, value similarity is based on the principle that the more data items referring to the same real-world event have nearby values, the higher the trust scores of these items are. We use the distribution of sensed values to increase trust scores for values near the mean of the distribution, and decrease them for outlier values.

Specifically, we consider that sensed values corresponding to the same event follow a Gaussian distribution D with mean μ and standard deviation σ. Then, the trust score for a data item d with value v_d is denoted as $\hat{s}_d$ and is calculated as follows:

$$\begin{aligned}\hat{s}_d &= 2\left(0.5 - \int_{\mu}^{v_d} f(x)\,dx\right) \\ &= 1 - \int_{2\mu - v_d}^{v_d} f(x)\,dx = 2\int_{v_d}^{\infty} f(x)\,dx\end{aligned} \tag{2}$$

Figure 2 shows how to compute the integral area for trust score s_d. In the figure, the shaded area represents the initial score of $\hat{s}_d$, which is in (0,1]. Here, the score $\hat{s}_d$ increases as the value v_d is closer to μ.

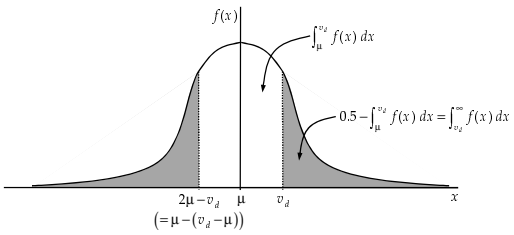


Fig. 2. Computing the intermediate score of $\hat{s}_d$.

This is a meaningful model because the mean μ is the most representative value that reflects well the value similarity.

Thus, we conclude that the mean has the highest trust score; if the value of a data item is close to the mean, its trust score is relatively high; if the value is far from the mean, its trust score is relatively low.

Finally, we adjust the node trust score of a sensor n based on the new trust scores of data items in D_n, where D_n includes the data items that originated at sensor node n. The formula for updating scores is as follows:

$$s_n = \frac{\sum_{d \in D_n} \hat{s}_d}{|D_n|} \tag{3}$$

For more details about trustworthiness measurement, please refer to our previous work in [12]. Even though we use the same trustworthiness assessment mechanism, we emphasize that our novel contribution in this work resides elsewhere, namely in the game-theoretic strategy of attacker-defender interaction. Furthermore, any alternative trust scores assignment mechanism that maps values to the $[0, 1]$ interval can be used in conjunction with our defense strategy.

C. Stackelberg Competitions

The *Stackelberg competition* concept represents a type of strategic games in economics, and it characterizes situations where two players compete for the same (bounded) market volume. Each player has its own utility, which is a function of its market share. Furthermore, the game is modeled as a *sequence* of actions, where one of the players initiates the game (the *leader*, denoted by L), whereas the other (the *follower*, denoted by F) responds with an action that is optimal for F, given the action of the leader. In the following, we present the detailed mathematical formulation of the Stackelberg competition model.

The utility (or benefit) for the leader and follower are denoted by the functions $U_L(A_L, A_F)$ and $U_F(A_L, A_F)$, where A_L and A_F are the actions taken by L and F, respectively. The Stackelberg model assumes that the leader has knowledge about the utility function of the follower, an assumption that is reasonable in many practical situations. For instance, in the case of cyber-terrorism against the power grid, the utility of an attacker is characterized by the amount of power stations that are rendered unfunctional during the attack. Consider a hypothetical action A_L^0 of the leader. Since L knows U_F, L will be able to determine what is the optimal action A_F^* of the follower, given that A_L^0 was executed. Specifically,

$$A_F^*(A_L^0) = \operatorname*{argmax}_{A_F} U_F(A_L^0, A_F) \tag{4}$$

Note that, the leader is able to determine the optimal action of the follower A_F^* (i.e., the only feasible action for the follower, assuming that the follower is acting rationally) before actually executing its own action A_L^0. As a consequence, the leader is able to determine its optimal action, by computing

$$A_L^* = \operatorname*{argmax}_{A_L^0} U_L(A_L^0, \operatorname*{argmax}_{A_F} U_F(A_L^0, A_F)) \tag{5}$$

That is, the value of A_L that maximizes the leader's utility after both the leader and the follower have executed their actions, in sequence. By choosing the appropriate A_L^*, the leader is capable of effectively limiting the actions of F, since a rational follower will only have the choice

$$A_F^* = \operatorname*{argmax}_{A_F} U_F(A_L^*, A_F) \tag{6}$$

The pair of actions (A_L^*, A_F^*) represent the *subgame perfect Nash equilibrium* of the game, and correspond to the optimal actions for both (rationally-behaving) players.

The main idea behind the Stackelberg competition model is that both the leader and the follower behave rationally, and the game will converge to the *subgame perfect Nash equilibrium* of the game, corresponding to the optimal actions for both players. The Stackelberg competition model has proven to be suitable for several security applications (e.g., protection of national borders and protection of critical infrastructures). In the security realm, the defender takes the role of the leader, whereas the attacker takes the role of the follower. The principles underlying the Stackelberg model are:

(S1) The leader has an advantage in taking the first action
(S2) The follower behaves rationally
(S3) The leader knows the utility function of the follower

In Section III, we discuss why the above principles fit the studied sensor network defense problem setting.

III. Defender-Attacker Game

To model the game of attack and defense in a sensor network with N nodes $n_1, \ldots, n_N$, we introduce the concepts of budgets and utilities. On the one hand, the attacker tries to maximize the effect of its attack (quantified as the deviation in the value accepted by the sink) given a budget constraint M. On the other hand, the defender has limited budget K and tries to accomplish the necessary amount of node protection to keep the system within nominal parameters (Section IV will describe two distinct representations of node protection). The budgets M and K are application-specific, but intuitively higher budgets incur higher costs.

An attack is performed by altering (increasing or decreasing) the value of sensor readings. The amount by which values are modified by attackers is proportional to the budget spent, but we assume that no individual reading may be altered by more than a ratio of σ compared to the original reading value. The assumption of the *maximum attack capability* σ is reasonable because if a reading deviates from the average by a large amount, it is automatically discarded as not truthful (e.g., indicative of device malfunction), regardless of the trustworthiness score of the generating node. The σ value represents the constraint on the attacker's capability, and in practice may be assigned the maximum sensing device error of any individual node (this is application specific). We formally define an attack strategy as follows:

Definition 1: An attack strategy AS(M, σ), where M is the attack budget and σ is the maximum capability of the attack, alters the genuine sensed value v to $v(1 \pm \sigma)$. An attack strategy is represented as a set of pairs (n_i, m_i) where m_i $(1 < i \leq N)$ is the cost spent to attack node n_i and $\sum_{i=1}^{N} m_i \leq M$. $\square$

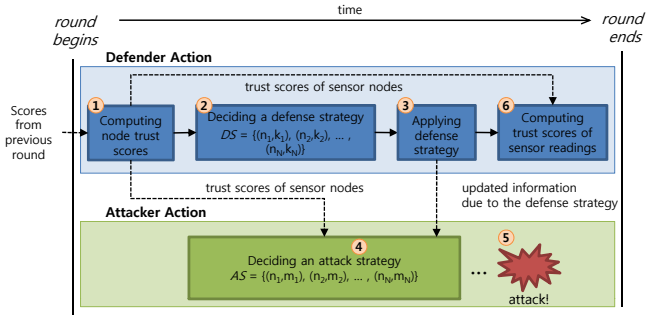


Fig. 3. The overall framework of the proposed defense-attack approach based on Stackelberg competitions.

The defender has the option to protect a set of sensor nodes to counter the attacker's strategy. If a node is protected, then the attacker may either be unable to compromise that node, or it must spend a higher cost to modify the corresponding value. We formally define a defense strategy as follows:

Definition 2: A defense strategy DS(K), where K is the defense budget, is specified as a set of pairs (n_i, k_i) where k_i $(1 < i \leq N)$ is the cost spent to protect n_i and $\sum_{i=1}^{N} k_i \leq K$. □

Note that, our model is generic and considers abstract representations of costs. In real-case scenarios, the cost of attack may be represented by the computational overhead required to break into a sensor node. Similarly, the cost of defense may quantify the overhead of activating physical perimeter protection around a sensor.

An attack is considered successful if the value accepted by the sink differs by at least a ratio of Δ from its genuine value:

Definition 3: An attack AS(M, σ) is successful against defense DS(K) if the deviation of the accepted value $Dev(K, M) \geq \Delta$, where Δ is a given threshold. Here, $Dev(K, M) = \frac{I_A}{I_O}$ where I_A is the value accepted by the sink and I_0 is the genuine value. □

Note that, this model is relevant in practical applications. For instance, in the case of a power grid, consider that the value of the electrical current in a network of monitored lines must be I_0. If the electrical current reading increases above a certain threshold $I_0 \cdot \Delta$ (e.g., $\Delta = 1.2$ means a 20% increase), the system is programmed to automatically decrease the generator output in compensation. However, if the genuine value of the current is actually I_0 and the attacker determines the sink to accept the wrong $I_0 \cdot \Delta$ value, then the new current value will become too low causing malfunction of the entire power grid.

Figure 3 shows the overview of the proposed defense-attack game within the Stackelberg competition [10] model. The actions take place on either the attacker's side or the defender's side. The solid lines represent action sequencing, whereas the dotted ones indicate the information flow between steps. For instance, information about the trust scores flows from step 1 towards step 4. Note that, in our model we consider the worst case scenario where the trust scores and the defense strategy are available to the attacker. For instance, a determined attacker may monitor communication and execute the same score-assigning procedure as the defender.

The time is discrete and evolves in rounds. In each round, the defender acts as the *leader* of the competition, whereas the attacker acts as the *follower*. Each round consists of six steps: In step 1, the trust scores of sensor nodes are calculated based on the input from the previous round, which consists of *(i)* trust scores of data items from the previous round, *(ii)* the value distribution of data items, and *(iii)* trust scores of sensor nodes in the previous round. In step 2, the defender (i.e., leader) determines the defense strategy $DS = \{(n_1, k_1), (n_2, k_2), ..., (n_N, k_N)\}$ to protect the sensor network. According to the Stackelberg model, we assume that the defender knows the attacker's maximum budget while deciding on DS. The goal of step 2 is for the defender to find the optimal DS that achieves the required protection amount and minimizes the defense cost $\sum_{i=1}^{N} k_i$. In step 3, the defender applies the strategy DS and updates the protection information of each node. In step 4, the attacker decides on attack strategy $AS = \{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\}$. The attacker's goal is to maximize the amount of distortion in the accepted value within the limited budget, i.e., $\sum_{i=1}^{N} m_i \leq M$. In step 5, the attacker alters the readings of attacked nodes according to *AS*. Following the attack, in step 6, the trust scores of data items are calculated based on the newly reported values. If the defense is successful, the value accepted by the server will not differ from the genuine value by more than a ratio of Δ.

The principles of Stackelberg competitions are abided by in the sensor network defense model as follows: (S1) holds because the defender has the incentive to protect the correct functionality of the network, and also the capability to strengthen sensor nodes before they can be attacked. (S2) holds because the optimal action of the attacker, i.e., the one that inflicts the most significant damage to the accepted value, is obtained only if the attacker behaves rationally (e.g., it focuses on less protected nodes). Finally, (S3) holds because the optimal utility of the attacker, i.e., the most efficient way for the attacker to bring down the system, is a deterministic function of node trust scores, hence it is known to the defender.

IV. Defense and Attack Strategies

We introduce two solutions for defense strategies in sensor networks. The first one addresses the case of a *binary protection* model, where a sensor node can be either fully protected, or fully unprotected. The defender will spend a budget of either 0 or 1 to protect a node, hence the "binary" term. If a node is protected, then any attack on that node will fail. The second case is the *fractional protection* model, and is more flexible, allowing the defender to allocate an arbitrary amount of budget to protect one node. An attacker may still be able to attack a protected node, but the effect of the attack will be reduced in proportion to the amount of node protection. We present the two models in Sections IV-A and IV-B, respectively.

A. Binary Protection Model

In this model, the cost for attack and defense is identical for all nodes, and the attacker/defense budgets M and K represent

the maximum number of nodes that can be attacked/protected in each round. When a node is protected, any attempted attack on it will fail. Hence, the reading of that node will be more trustworthy, and its trust score is increased by the defender by a factor of ρ compared to its current score. For example, if $\rho = 1.5$, the trust score of each protected node increases by 50% (note that, the resulting score is truncated to the maximum value 1 if the product of multiplication with ρ is greater than 1).

According to Equation 1, the value accepted by the sink is obtained as the mean of the individual readings weighted by node trust scores. Therefore, it is easy to prove that the optimal strategy for the defender is to protect the nodes with the highest trust scores. This way, even if the attacker compromises nodes with lower trust scores, the impact of the attack will be low. Assume that nodes are sorted in decreasing order according to trust scores. Then the defender will protect a number of k nodes $n_1, \ldots, n_k$ ($k \leq K$).

The attacker, who in the worst case is assumed to have complete knowledge about trust scores and defense budget, will act rationally, and will attempt to attack nodes with as high trust scores as possible. However, s/he will not attack any of the protected nodes, since doing so will bring no benefit to the attack. (Note that, in the general case when an adversary has less knowledge than in the worst case, such an outcome is possible, leading to decreased attack impact. However, we address only the worst case which guarantees network protection in all other cases as well).

The defender will proceed to compute the optimal defense strategy, i.e., the k value that ensures that an attack is not successful according to Definition 3. To that extent, the defender (which acts as the game leader) will determine the subgame perfect Nash equilibrium (A_L^*, A_F^*) (see Section II-C). The defender considers its own hypothetical action $A_L^0 = k$, and determines that the only rational action of the follower (i.e., the attacker) is to attack the set of nodes $X(k) = n_{k+1}, \ldots, n_{k+M}$, i.e., unprotected nodes with highest trust scores. Denote the attacker's action by $A_F^*(k) = X(k)$ (note that, it is possible that $|X| < M$, in the case when $k > N - M$). The value I_A accepted by the server[1] after the attack is:

$$\frac{\sum_{j=1}^{k} v_j \cdot s_j \cdot \rho + \sum_{j=k+1}^{k+1+M} v_j \cdot s_j \cdot (1+\sigma) + \sum_{i=k+M+2}^{N} v_j \cdot s_j}{\sum_{j=1}^{k} s_j \cdot \rho + \sum_{j=k+1}^{N} s_j}$$

According to Definition 3, the objective of the defender is to keep the system working correctly, therefore it must hold that $Dev(K, M) = I_A/I_0 < \Delta$. On the other hand, the cost bared by the defender grows larger as k increases. To compute the subgame Nash equilibrium state, the defender must determine the minimum value k for which the above condition holds, i.e.,

$$A_D^* \equiv k^* = \{k_0 | (Dev(k_0, M) < \Delta) \wedge (\forall k < k_0, Dev(k, M) \geq \Delta)\}$$

It can be easily observed that $Dev(k, M)$ is a monotonically decreasing function of k, due to the fact that the node sequence is sorted in decreasing order of trustworthiness scores. As a result, the defender can efficiently find the value of k^* by performing a binary search in the interval $\{1, \ldots, N\}$.

B. Fractional Protection Model

The binary model is simple and intuitive, but relies on the strong assumption that a node that is protected is immune to attacks. We relinquish this assumption with the fractional protection model, whereby attacking a protected node has an effect that depends on the node's *degree of protection*. The degree of protection captures both the amount by which the value reported by a node can be altered by the attacker, as well as the cost that must be paid by the defender to increase protection. The degree of protection of sensor node n_i is denoted by p_i and has values in the interval $[0, 1]$, where 0 signifies no protection and 1 signifies perfect protection. When an attacker spends a budget m ($\leq M$) to attack n_i, the amount of change in reported value is captured by the *attack effect* $ae_{i,m}$ defined as follows:

$$ae_{i,m} = \min\ ((1 - p_i) \cdot m \cdot \sigma,\ \sigma) \tag{7}$$

Note that, the attack effect increases with the cost spent, but it can not exceed maximum capability σ. In the worst case, the attacker will know the protection degrees, and will not waste the attack budget beyond $\sigma/(1-p_i)$. The sensed value v_i from node n_i is modified to $v_i + v_i \cdot ae_{i,m}$.

The cost required for additional protection also varies with the degree of protection. Intuitively, if a node is already highly protected, the cost required to increase the protection towards the ideal value 1 increases more than linear in the amount of additional protection. Given node n_i with protection degree p_i, the increase in protection obtained with k units of defense budget (i.e., to increase the protection degree from p_i to $p_i + de_{i,k}$) is

$$de_{i,k} = \sum_{j=1}^{k} \frac{(1 - p_i)}{2^j} = (1 - p_i) \cdot (1 - \frac{1}{2^k}) \tag{8}$$

Equation 8 signifies that the cost for protecting n_i increases exponentially as p_i approaches 1. In the following, we describe the attack and defense strategies that lead to Nash equilibrium.

Attacker Strategy. The attacker must choose the strategy $AS(M, \sigma) = \{(n_1, m_1), (n_2, m_2), ..., (n_N, m_N)\}$ which maximizes the deviation in the value accepted by the server. Note that, there are two components that need to be considered: the attack effects according to Equation 7, as well as the trust score of each node, since trust scores affect accepted values according to Equation 1. Specifically the attacker should choose to attack first a sensor node n_i whose trust score s_i is

[1]Without loss of generality, we assume that all malicious value deviations occur in the same direction. This is the worst case, since if some are positive and other negative, their effects will cancel out.

Algorithm 1 *AttackStrategy*

for each sensor node n_i, compute Equation 9;
Sort nodes in decreasing order of the value computed in step 1;
$tc = 0$; /* total cost */
for i = 1 to N
  $m_i = \frac{1}{1-p_i}$;
  $tc = tc + m_i$;
  if $tc \geq M$
    then m_i = m_i - (tc - M);
      return the pairs of (n_j, m_j) for $1 \leq j \leq N$;
  end if
end for
return the pairs of (n_i, m_i) for $1 \leq i \leq N$;

high, but its protection degree p_i is low. The contribution of n_i in altering the accepted value is

$$(1 - p_i) \cdot s_i \cdot m_i \cdot \sigma \tag{9}$$

In the optimal strategy, attackers first sort the sensor nodes in decreasing order according to Equation 9, and attack the first N' nodes with individual budgets $m_i = \frac{1}{1-p_i}$ (since $(1 - p_i) \cdot m \cdot \sigma = \sigma$) such that $\sum_{i=1}^{N'} m_i \leq M$. The attacker optimal strategy is summarized in Algorithm 1. The time complexity of finding the strategy is O(N log N).

The following theorem shows that *AttackStrategy* in Algorithm 1 is optimal.

Theorem 1: Let the set of sensor nodes be ordered decreasingly according to Equation 9, and let v_O be the altered value accepted by the sink after an attack with (optimal) strategy AS_O determined according to Algorithm 1. Then there exists no alternative attack strategy AS_A that determines the sink to accept a value v_A such that[2] $v_O < v_A$.

Proof: Informally, the optimality means that there is no other alternative attack strategy AS_A that can make more distortion in the accepted value (i.e., weighted mean) compared to the attack strategy AS_O generated by the algorithm. To show this, we first illustrate the result of the algorithm, AS_O, as follows:

$AS_O = \{m_1, m_2, m_3, ..., m_l, m_{l+1}, ..., m_N\}$[3], $\sum_{i=1}^{N} m_i = M$

$$m_i = \begin{cases} \frac{1}{1-p_i} & \text{for } 1 \leq i \leq l \\ M - \sum_{j=1}^{l} m_j & \text{for } i = l+1 \\ 0 & \text{for } l+2 \leq i \leq N \end{cases}$$

The result shows that the algorithm assigns costs for the first l nodes to achieve the maximum capability attack effects (i.e., "full" attack; assigning more budget to any such node would not make sense, as the capability would be exceeded). The $(l+1)^{th}$ node has the remainder from the budget M (i.e., "partial" attack), and the other nodes do not have any attack budget assigned (i.e., no attack). Figure 4 shows the assigned attack budget for each node in the graph of $ae_{i,m}$ versus m. In AS_O, the first l nodes are located in point A in the graph, one node is located between points A and B, and the other nodes are located in B.

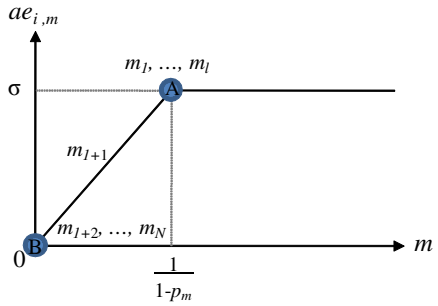


Fig. 4. A graph of $ae_{i,m}$ versus m.

To prove the optimality of AS_O, we first define the cost redistribution from m_i to m_j ($i \neq j$) in an attack strategy AS as follows:

Definition 4: For an attack strategy $\{m_1, m_2, m_3, ..., m_i, ..., m_j, ..., m_N\}$, a cost distribution from m_i to m_j ($i \neq j$) with the amount of cost Δ_m ($\Delta_m \leq m_i$) is denoted as $T_{i,j}^{\Delta_m}$ and changes the attack strategy into $\{m_1, m_2, m_3, ..., m_i - \Delta_m, ..., m_j + \Delta_m, ..., m_N\}$. □

Informally, the cost redistribution signifies that a budget Δ_m is subtracted from m_i and added to m_j, all other costs being left unchanged.

We can see that any alternative attack strategy AS_A can be expressed as a sequence of $N - 1$ cost redistributions starting from AS_O (denoted as $T_{1,2}^{\Delta_1}$, $T_{2,3}^{\Delta_2}$, ... $T_{N-1,N}^{\Delta_{N-1}}$) since the attack budget of the first l nodes is $\frac{1}{1-p_i}$ and it is the maximum that can be allocated to those nodes (full attack). More formally, an alternative attack strategy $AS_A = \{m'_1, m'_2, m'_3, ..., m'_N\}$, $\sum_{i=1}^{N} m'_i = M$ is represented as the cost redistributions from the optimal attack strategy, $AS_O = \{m_1, m_2, m_3, ..., m_l, m_{l+1}, ..., m_N\}$, and the cost redistributions are denoted as $T_{1,2}^{m_1 - m'_1}$, $T_{2,3}^{(m_1 - m'_1) + (m_2 - m'_2)}$, ... , $T_{N-1,N}^{\sum_{i=1}^{N-1}(m_i - m'_i)}$. Note that, the cost redistribution among m_1, ..., m_l does not provide any additional benefit since the attack effect $ae(i, m_i)$ for $0 \leq i \leq l$ is already σ (i.e., maximum strength of the attack). Therefore, we only consider the cost redistribution from $\{m_1$ ... $m_{l+1}\}$ to $\{m_{l+1}$ ... $m_N\}$. Figure 5 shows the chained cost redistribution from AS_O to AS_A.

By using the definition of the cost redistribution, we prove the optimality of AS_O as follows:

Denote the sequence of cost redistributions from AS_A to AS_O as $T(1)$, $T(2)$, ..., $T(N-1)$. We prove the theorem by mathematical induction for the q-th redistribution.

Initial Step ($q = 1$): For $T(1)$, consider the decrease in the

[2]We assume the worst case when all deviations are positive. The case of negative deviations is analogous.

[3]Here, we assume that the nodes are sorted in descending order of the attack efficiency (i.e., Equation 9) and omit the node numbers for simplicity.

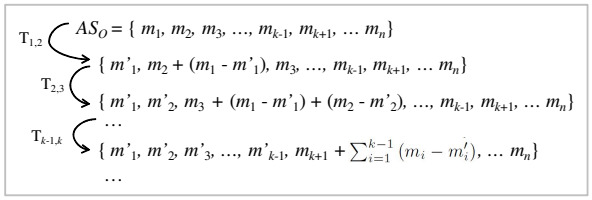


Fig. 5. Chained cost redistributions from AS_O to AS_A.

accepted value due to the first node, $D = v_1 \cdot (1 - p_1) \cdot \sigma \cdot (m_1 - m'_1) \cdot s_1 \; / \; \sum_{l=1}^{N} s_l$ and the increase due to the second node, $I = v_2 \cdot (1 - p_2) \cdot \sigma \cdot (m_1 - m'_1) \cdot s_2 \; / \; \sum_{l=1}^{N} s_l$. To satisfy $v_{AS_O} < v_{AS_A}$, $D < I$ should be true (the amount of increase should be larger than the amount of decrease). But we know that $D < I$ is false since by construction the nodes are sorted in the decreasing order of the attack effect, that is, $(1 - p_i) \cdot \sigma \cdot s_i > (1 - p_j) \cdot \sigma \cdot s_j$ is always true where $i < j$. Therefore, the theorem is true when q = 1.

Inductive Step: For all $q < N$, We assume that the theorem is true for $T(q-1)$ which denotes the cost redistribution $\{m'_1, m'_2, m'_3, ..., \; m_{k-1} - \sum_{i=1}^{q-1}(m_i - m'_i), m_q + \sum_{i=1}^{q-1}(m_i - m'_i), ..., m_N\}$. This implies $(1 - p_{k-1}) \cdot \sigma \cdot \sum_{i=1}^{q-1}(m_i - m'_i) \cdot s_{q-1} > (1 - p_q) \cdot \sigma \cdot \sum_{i=1}^{q-1}(m_i - m'_i) \cdot s_q$. Then, the theorem is also true for $T(q)$ since $(1 - p_q) \cdot \sigma \cdot \sum_{i=1}^{q}(m_i - m'_i) \cdot s_q > (1 - p_{q+1}) \cdot \sigma \cdot \sum_{i=1}^{q}(m_i - m'_i) \cdot s_{q+1}$ is true based on the assumption, $(1 - p_q) \cdot \sigma \cdot s_q > (1 - p_{q+1}) \cdot \sigma \cdot s_{q+1}$.

Therefore, there is no AS_A which satisfies $v_{AS_O} < v_{AS_A}$. ■

Defender Strategy. According to the Stackelberg competition model, the defender is aware of the optimal strategy of the attacker (Algorithm 1), and will devise a defense strategy within allocated budget K in order to prevent the attacker from being successful. Recall that, the defense strategy is represented as $DS = \{(n_1, k_1), (n_2, k_2), ..., (n_N, k_N)\}$ where $\sum_{i=1}^{N} k_i \leq K$. The objective of the defender is to maximize the amount of node protection while minimizing protection cost.

According to Definition 3, in order to prevent the attack from succeeding, the defender must ensure that the following condition holds:

$$\frac{I_A}{I_0} < \Delta$$

which is equivalent to

$$\frac{\frac{\sum_{i=1}^{N} s_i \cdot (v_i + v_i \cdot ae_{i,m_i})}{\sum_{i=1}^{N} s_i}}{\frac{\sum_{i=1}^{N} s_i \cdot v_i}{\sum_{i=1}^{N} s_i}} < \Delta$$

and further,

$$\sum_{i=1}^{N} s_i \cdot v_i \cdot ae_{i,m_i} < (\Delta - 1) \cdot \sum_{i=1}^{N} s_i \cdot v_i$$

Given that all sensed values will have to be located within an interval of $\pm\Delta$ from some genuine value, we simplify the equation by assuming sensibly equal v_i values, and we obtain the following condition that should be satisfied:

$$\sum_{i=1}^{N} s_i \cdot ae_{i,m_i} < (\Delta - 1) \cdot \sum_{i=1}^{N} s_i \tag{10}$$

The right-hand side in the above inequality, $(\Delta - 1) \cdot \sum_{i=1}^{N} s_i$, is a constant for a given round, whereas $\sum_{i=1}^{N} s_i \cdot ae_{i,m_i}$ is the optimal allocation of budgets by the attacker, which is known to the defender. The defender will counter the attack by performing a set of protection operations, captured by the allocation of the defense budget to nodes.

The most effective way for the defender to minimize the effect of the attack is to increase the degree of protection for sensor nodes who already have high trust scores, but their degrees of protection are not very high. The second condition results as a consequence of the defense effect (Equation 8), since it is very expensive to protect nodes that already have high degrees of protection. Specifically, nodes will be chosen for protection according to a criterion given by the following expression:

$$(1 - p_i) \cdot s_i \tag{11}$$

Equation 11 provides the order in which nodes will be protected. The protection budgets k_i are determined in an incremental fashion, as described in Algorithm 2. The node n_i for which the value in Equation 11 is highest is assigned a budget of 1, then the protection degree is re-computed, and Equation 11 is re-evaluated. The process stops when the condition from Equation 10 is satisfied. The time complexity of algorithm *DefenseStrategy* is O($N \log N$) since the algorithm needs to sort N sensor nodes. Furthermore, the incremental allocation of defense budgets takes $O(K \log N)$ if an efficient priority queue structure is used.

Algorithm 2 *DefenseStrategy*

```
1: K = 0;
2: for each sensor node n_i compute defense effect
       according to Eq. 11;
3: Sort nodes in a heap in decreasing order of the
       value computed in step 2;
4: while (1)
5:     Choose n_1 (top of heap) and increase k_1 by 1;
6:     K = K + 1;
7:     Update p_1 and re-insert n_1 in the sequence;
8:     ta = Σ_{i=1}^{N} s_i · ae_{i,m_i} /*update total attack effect*/
9:     if ta < (Δ − 1) · Σ_{i=1}^{N} s_i
10:        then return the pairs of (n_i, k_i) for 1 ≤ i ≤ N;
11:    end if
12: end while
```

V. Experimental Evaluation

In this section, we present the results of our performance evaluation. In Section V-A we outline the experimental testbed settings. Next, in Section V-B we evaluate the *effectiveness* of the proposed defense strategies. Finally, in Section V-C we

TABLE I
SUMMARY OF NOTATIONS.

Symbol	Definition	Default
N	number of sensor nodes	1,500
v	genuine value	100
ϵ	error rate	0.1
σ	maximum attack capability	2.0
ρ	strengthening for protected nodes	1.1
Δ	attack success/fail threshold	1.2
M	attacker's budget	600

measure their *efficiency*, i.e., the overhead incurred in the online phase of strategy computation.

A. Experimental Settings

We simulate a sensor network where nodes sense process parameters and send their readings to a sink node. Since our defense strategy works at the data layer, we are not concerned with the topology of the network or routing issues (as mentioned before, we assume that a compromised node can modify/generate only the data items collected by that node). We vary the number of sensor nodes N between 500 and $2,500$, and we consider observations that span over a number of up to 100 rounds. Initially (i.e., at round 0), we set the trust score of each sensor node to 0.5 (which is also the initial trust score of data from each node). All the experiments have been conducted on an Intel 2.2GHz Core2 Duo processor with 2GB RAM running Windows 7. The prototype is developed in Java (JDK 1.6.0). Table I summarizes the experimental parameters and their default values.

Our workload consists of synthetic data with a single attribute (the sensed parameter, i.e., electric current). Sensor readings are determined as follows: first, a genuine value v is generated, which represents the actual process value. Note that, this value is not known exactly to neither the nodes, nor the sink, even in the absence of attacks, due to measurement errors. In absence of attacks, given an error range ϵ, the simulation assigns to each node a reading uniformly generated at random between $v - v \cdot \epsilon$ and $v + v \cdot \epsilon$. The range ϵ is the equivalent of device error tolerance.

We consider two types of attacks:

- Random attack: Adversaries randomly choose sensor nodes to be attacked, as well as the attack budget spent on each node. This type of attack is applicable to cases where adversaries do not have any knowledge (or have incomplete knowledge) about the sensor node trust scores and degrees of protection.
- Worst case attack: Adversaries choose the set of attacked nodes and the corresponding attack budget for each node based on the optimal strategy described in Section IV. This type of attacks captures the case where adversaries have full knowledge about the sensor network and the defender's strategy.

For the fractional protection model, we consider two specific types of attacks which fall between the random and worst-case attacks in terms of effectiveness. The adversaries only have partial knowledge about the sensor network and the defender's strategy, as follows:

- Maximum trust score attack: Adversaries choose sensor nodes in decreasing order of the node trust scores. Protection degrees are not factored in the attack decision.
- Minimum degree of protection attack: Adversaries choose sensor nodes in increasing order of nodes' protection degrees. In other words, the least protected sensor nodes are attacked first. Node trust scores are not factored in the decision.

The rationale behind considering several types of attacks (other than the ones considered in Section IV) is to show the versatility of the proposed defense strategies, as well as their applicability to a broad range of scenarios.

B. Effectiveness in Defending against Attacks

We first present an experiment that motivates the need for effective protection strategies in sensor networks. Figure 6 shows the accepted values in the case of random attacks and in the absence of any protection by the defender (i.e., trust scores are used to calculate the accepted value without protecting any nodes). Observe that, the accepted values exceed the tolerable error boundary (i.e., Δ = 1.2). Even when only 40% of the 1500 sensor nodes are attacked with the attack capability $\sigma = 2.0$, the attack is successful since the accepted value is distorted more than Δ. These results show that relying on data trustworthiness scores alone, without any node protection, is not sufficient to keep the accepted values within the nominal bounds.

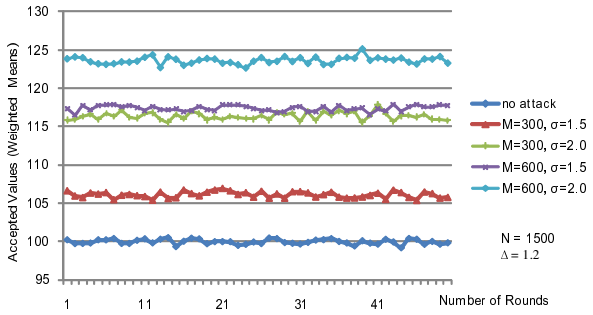


Fig. 6. Accepted value in the presence of attacks, without protection (initial score = 0.5, ϵ = 0.1, genuine value v = 100).

Next, we show how the proposed defense strategies manage to protect against attacks. Figure 7 shows the accepted values under worst case and random attacks in the binary protection model. Each dot represents the accepted value in a particular round. We can observe that accepted values are always within the acceptable threshold Δ = 1.2 regardless of the attack type. For the worst case attack, accepted values are just below the threshold Δ, since this threshold is used as reference point both by adversaries trying to maximize the attack effect, as well as by the defense strategy that attempts to minimize the defense cost.

Next, in Figure 8 we show the accepted values under the four types of attacks in the fractional model. Similar to the binary model, we can observe that accepted values are

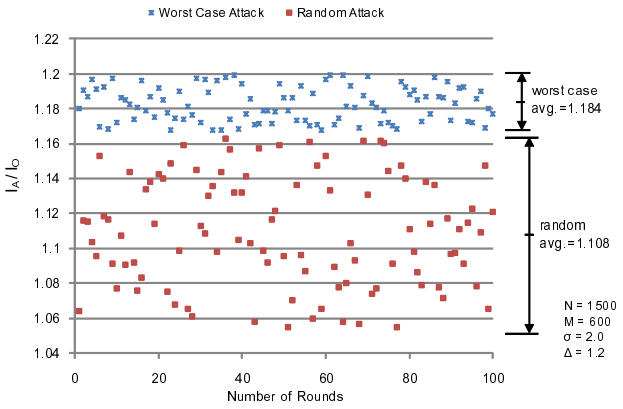
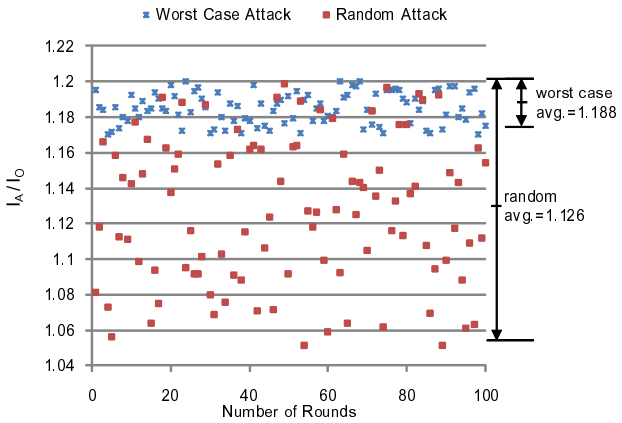
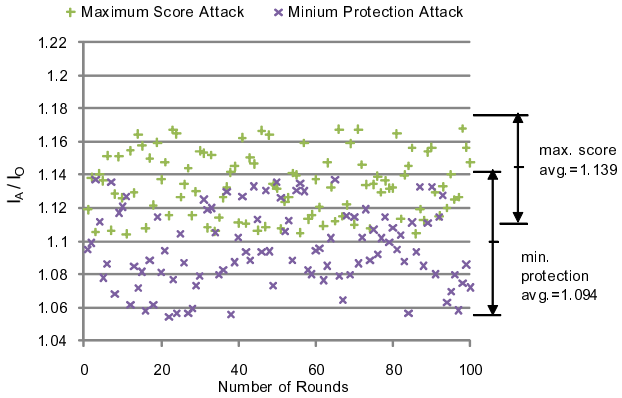


Fig. 7. Distortion of the accepted value (i.e., $\frac{I_A}{I_0}$) under attacks in the binary model.



(a) Worst case attack and random attack



(b) Max. score attack and min. protection attack

Fig. 8. Distortion of the accepted value (i.e., $\frac{I_A}{I_0}$) under attacks in the fractional model

always within the acceptable threshold $\Delta = 1.2$ regardless of the attack type. For the maximum score attack and the minimum protection attack (Figure 8(b)), we can see that the effectiveness of these attacks is much lower than that of the worst case attack.

Next, we measure the changes of defense cost when we run several rounds of the defense-attack game. The purpose of this experiment is to show how the proposed defense strategy adapts to attacks, and is able to maintain the defense cost below the maximum budget whenever possible. Figure 9 shows the trend of the defense cost when running the defense strategy with different attack capabilities (i.e., σ). For the binary protection model (Figure 9(a)), we also vary the ratio of increasing the trust scores for protected nodes (i.e., ρ). To quantify defense cost, we measure the percentage of protected nodes for the binary model, and the average degree of protection for the fractional model.

The results show that our approach achieves protection with reasonable cost. In the binary model (Figure 9(a)), we can observe that the number of protected nodes decreases drastically just after the first few rounds, and remains stable. The reason is that in the binary model, the protected nodes are continuously strengthened and thus have higher trust scores as rounds progress. Therefore, the defender can achieve high data trustworthiness with a relatively small number of protected nodes (around 30% for $\sigma = 1.5$ and $\rho = 1.4$). In the fractional model (Figure 9(b)), we can see that the average degree of protection decreases slightly and remains stable after several rounds. The reason is that the attack and the defense reach a balance due to the trade-off between trust scores and degree of protections. From the results, we can see that after 50 rounds, less than 65% degree of protection on average per node is needed to defeat the worst case attack.

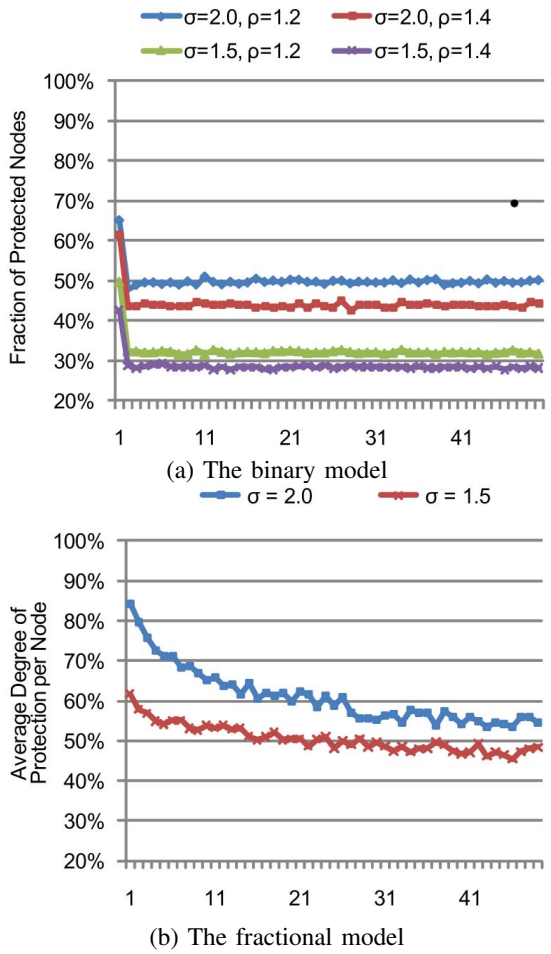
C. Efficiency of On-line Strategy Computation

To evaluate defense strategy computation efficiency, we measure the elapsed time required to decide which nodes are to be protected, and by what protection amounts. It is important to minimize the elapsed time since in sensor network environments, the decision should be done in close to real time. Figure 10 shows the elapsed times when the number of nodes (N) and the attack budget (M) vary.

In the binary model (Figure 10(a)), we can see that the performance scales well as the number of sensor nodes increases (as $N \log N$). Also, we can observe that the elapsed time is less affected by the attack budget. The reason is that the search space for deciding the optimal strategy only depends mostly on the number of nodes, and less on the attacker's capability or budget. In the fractional model (Figure 10(b)), we can see that the elapsed time is higher than in the binary model, but still low in absolute value (less than 1 second for 2,500 sensor nodes). Furthermore, the method is scalable when the number of nodes increases (due to the $N \log N$ dependence).

VI. Related Work

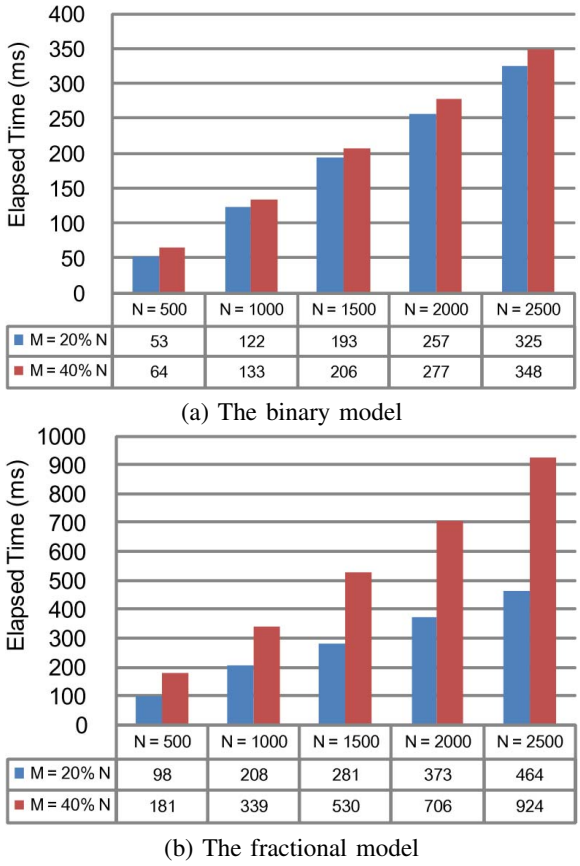
Data trustworthiness is a serious concern for professionals involved in a wide array of information systems, ranging from data warehousing to supply chain management. One industry study estimated the total cost to the US economy due to data trustworthiness problems at over US$600 billion



(a) The binary model

(b) The fractional model

Fig. 9. Trends of the defense budget under attack (initial scores = 0.5, first 50 rounds).



(a) The binary model

(b) The fractional model

Fig. 10. Execution time(ms) for deciding a defense strategy for a round.

per annum [6]. Still, few solutions are available today for sensor network data trustworthiness assessment. One such work [20] studies data trustworthiness for sensor networks in cyber-physical systems, where the purpose of the sensors is to detect enemy presence in the battlefield. The goal of the trustworthiness analysis is to eliminate false alarms due to noise in the environment or faulty sensors. The authors construct a graph in which sensors that should observe similar events are connected to each other, and an event is validated as an alarm if the readings from several connected sensors concur. However, [20] does not deal with malicious attacks, and there is no notion of protection of the critical infrastructure: only event filtering is addressed. Another data filtering technique is proposed in [7], where a large number of sensor networks provide scientific data about natural phenomena. A reputation-based mechanism is used to detect data that are not trustworthy. However, such a mechanism reacts slowly over time, and detects problems due mainly to inaccurate sensors, but does not deal with coordinated attacks where a significant fraction of the nodes are compromised by a malicious attacker.

Recently, security issues in sensor networks have received a lot of interest. The majority of existing research focuses on efficient query processing, secure data delivery, and secure routing protocols. A secure protocol for sensor networks was proposed by Perrig et al [19]. The protocol is based on authentication between sensors with a shared secret key. Luk et al. [15] propose a secure sensor network communication architecture which provides a high level of security while requiring low energy consumption. The authors employ a block cipher mode which provides both secrecy and authenticity in only one pass over the data. They exploit both unicast and broadcast modes to balance energy and communication efficiency. Ma et al. [16] provide an adversarial model and a secure data sharing method in unattached sensor networks which accumulate sensed data in sensors until an itinerant data collector consumes the data. However, none of the existing work addresses the issue of data trustworthiness assurance in adversarial environments.

Closer to our work, Liu et al [14] study attacks that inject false data into electric power grid networks. Similar in concept to the example we presented in Figure 1, they show that existing data filtering techniques only work against random faults, but not against a targeted attack. A formal procedure for circumventing statistical fault detection techniques is detailed, but no defense is provided. We consider the same kind of

adversarial scenario, but in addition we present a game-theoretic approach to defending the network against false data injection.

Due to strategic interactions between nodes in sensor networks, game-theoretic models have been applied to various sensor network problems including routing and security (see the survey by Machado et al. [17] for details). Among these models, the ones used for security are directly related to our work. In [3], a zero-sum game theoretical model is proposed to analyze situations in which the goal is to detect attacks on the most vulnerable node. In [11], a minmax optimal solution is given to detect malicious packets by sampling. In [18], a game-theoretic model for preventing denial of service attacks is given. In [2], denial of service in the routing layer is addressed using repeated games. Similarly, in [4], the authors consider a dynamic wireless sensor network of mobile nodes and develop a model that takes into account cooperation, reputation, and security. To the best of our knowledge, no existing work addresses the issue of sensor data trustworthiness in a game theoretical setting.

VII. Conclusion

In this paper, we addressed the important and challenging problem of assuring trustworthiness of sensor data in the presence of malicious adversaries. We developed a game-theoretic defense strategy to protect sensor nodes from attacks and to guarantee a high level of trustworthiness for sensed data. The objective of the defense strategy is to ensure that sufficient sensor nodes are protected in each attack/defense round such that the discrepancy between the value accepted by the sink and the truthful sensed value is below a certain threshold. We modeled the attack-defense interaction as a Stackelberg competition and provided two alternate utility and cost measures for the game. We implemented a prototype of the proposed strategies and showed through extensive experiments that our solution provides an effective and efficient way of assuring sensor data trustworthiness. In future work, we will investigate how to extend the defense model to other scenarios where value readings from different sensors are correlated through more complex functions (rather than equality). We also plan to deploy the proposed defense strategies in a real sensor network.

Acknowledgments: The work reported in this paper has been partially supported by NSF under grants CNS-1111512 and CNS-0964294.

References

[1] U.S.-Canada Power System Outage Task Force. Final report on the August 14, 2003 blackout in the United States and Canada. https://reports.energy.gov/B-F-Web-Part1.pdf, April 2004.

[2] A. Agah, M. Asadi, and S. Das. Prevention of DoS attack in sensor networks using repeated game theory. In *Proc. of the Int'l Conf. on Wireless Networks*, pp. 29–36, Citeseer, 2006.

[3] A. Agah, K. Basu, and S. Das. Preventing DoS attack in sensor networks: a game theoretic approach. In *Proc. of IEEE Int'l Conf. on Communications*, volume 5, pp. 3218–3222, 2005.

[4] A. Agah, S. Das, and K. Basu. A game theory based approach for security in wireless sensor networks. In *Proc. 2004 IEEE Int'l Conf. on Performance, Computing, and Communications*, pp. 259–263, 2004.

[5] E. N. Asada, A.V. Garcia and R. Romero. Identifying multiple interacting bad data in power system state estimation. IEEE Power Engineering Society General Meeting, pp. 571-577, 2005.

[6] W. Eckerson. Data Warehousing Special Report: Data quality and the bottom line. *Applications Development Trends May*, 2002.

[7] S. Ganeriwal, L. Balzano, M. B. Srivastava, Reputation-based Framework for High Integrity Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)* , May 2008.

[8] J. Fernandez and A. Fernandez. SCADA systems: vulnerabilities and remediation. *Journal of Computing Sciences in Colleges*, 20(4):160–168, 2005.

[9] S. Gastoni, G.P. Granelli and M. Montagna. Multiple bad data processing by genetic algorithms. *IEEE Power Tech Conference*, pp. 1-6, 2003.

[10] R. Gibbons. *A primer in game theory*. Harvester Wheatsheaf New York, 1994.

[11] M. Kodialam and T. Lakshman. Detecting network intrusions via sampling: A game theoretic approach. In *IEEE INFOCOM 2003. Twenty-Second Annual Joint Conf. of the IEEE Computer and Communications Societies*, volume 3.

[12] H.-S. Lim, Y.-S. Moon, and E. Bertino. Provenance-based trustworthiness assessment in sensor networks.. In *Proc. of the 7th Workshop on Data Management for Sensor Networks*, pp. 2–7. Singapore, 2010.

[13] A. Liu and P. Ning. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proc. of the 7th Int'l Conf. on Information Processing in Sensor Networks*, pp. 245–256. IEEE Computer Society Washington, DC, USA, 2008.

[14] Y. Liu, P. Ning and M. Reiter. False Data Injection Attacks against State Estimation in Electric Power Grids. *Proc. of the 16th ACM Conference on Computer and Communications Security (CCS)*, pp. 21-32, November 2009.

[15] M. Luk, G. Mezzour, A. Perrig, and V. Gligor. Minisec: a secure sensor network communication architecture. In *Proc. of the 6th Int'l Symposium on Information Processing in Sensor Networks*, pp. 479–488. ACM, 2007.

[16] D. Ma, C. Soriente, and G. Tsudik. New adversary and new threats: security in unattended sensor networks. *Network, IEEE*, 23(2):43–48, 2009.

[17] R. Machado and S. Tekinay. A survey of game-theoretic approaches in wireless sensor networks. *Computer Networks*, 52(16):3047–3061, 2008.

[18] J. McCune, E. Shi, A. Perrig, and M. Reiter. Detection of denial-of-message attacks on sensor network broadcasts. In Proc. of *2005 IEEE Symposium on Security and Privacy*, pp. 64–78, 2005.

[19] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler. SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.

[20] L.-A. Tang, X. Yu, S. Kim, J. Han, C.-C. Hung and W.-C. Peng Tru-Alarm: Trustworthiness Analysis of Sensor Networks in Cyber-Physical Systems. In *Proc. of Intl. Conference on Data Mining (ICDM),* pp 1079-1084, Sydney, Australia, December 2010.

[21] A. Wood and B. Wollenberg. *Power generation, operation, and control.* John Wiley and Sons, 2nd edition, 1996.