2007

A false injection-resilient scheme to monitor time-variant phenomenon in wireless sensor networks

Vinod Shukla
Iowa State University

Recommended Citation

Shukla, Vinod, "A false injection-resilient scheme to monitor time-variant phenomenon in wireless sensor networks" (2007). *Retrospective Theses and Dissertations*. 14653.
https://lib.dr.iastate.edu/rtd/14653

A false injection-resilient scheme to monitor time-variant phenomenon in wireless sensor networks

by

Vinod Shukla

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Daji Qiao, Major Professor
Yong Guan
Wensheng Zhang

Iowa State University

Ames, Iowa

2007

UMI Number: 1447503

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

DEDICATION

To my parents, my family and my friend, Živilė.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I would like to thank my adviser Dr. Daji Qiao for his valuable guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have greatly inspired me and renewed my hopes for completing my graduate education. Words are not enough to thank him for his constant support through the thick and thin of my stay at Iowa State University, and above all, for teaching me the way to do research.

I would also like to thank my committee members Dr. Yong Guan and Dr. Wensheng Zhang for their encouragement and support throughout my graduate education.

I would like to thank Dr. Arun Somani for his guidance throughout the initial stages of my graduate career.

I would like to thank my lab mates Wei, Guanqun and Xi who have gladly extended their help and support whenever needed. I would also like to thank my other friends at Iowa State with whom the two years became more memorable.

Deep appreciation and thanks to my friend and roommate Živilė for her support and care during the writing of my thesis. At last, I want to express my hearty gratitude to my parents, Mr. Motiram Shukla and Mrs. Madhuri Shukla and my brother and sister. Their faith in me has encouraged me to go after my dreams. Their constant motivation has made me work for those dreams and their selfless love and support seen me through failures and successes that have come along the way.

ABSTRACT

Being a promising technology which is envisioned to pervade numerous aspects of human life, wireless sensor networks are attracting remarkable attention in research community. The typical wireless sensors are small low-power, resource-constrained devices subject to functional failures which could be due to power loss or even malicious attacks on the devices. As the projected applications for wireless sensor networks range from smart applications such as traffic monitoring to critical military applications such as measuring levels of gas concentration in battle fields, security in sensor networks becomes a prime concern. In sensitive applications, it becomes imperative to continuously monitor the transient state of the system rather than steady state observations and take requisite preventive and corrective actions, if necessary. Also, the network is prone to attack by adversaries who intend to disrupt the functioning of the system by compromising the sensor nodes and injecting false data into the network. So it is important to shield the sensor network from false data injection attacks. Through this work, we prove that in the presence of adversaries, it would be difficult to correctly observe the transient phenomenon if sensors report just their readings. We develop a novel robust statistical framework to monitor correctly the transient phenomenon while limiting the impact of false data injection. In this framework, each sensor does a lightweight computation and reports a statistical digest in addition to the current sensed reading. Through a series of carefully-designed inter-sensor statistical tests on both the readings and digests, we are able to achieve our goal of preserving the transient phenomenon. We show a concrete realization of our statistical framework by developing a secure statistical scheme, called SSTF, to effectively monitor the transient phenomenon while being immune to false data injection attacks. SSTF is a two-tier system and the kernel of SSTF is our statistical framework, which is employed atop an enhanced version of the IHHAS security scheme. We present detailed theoretical analysis and in-depth simulation results to demonstrate the effectiveness of SSTF.

CHAPTER 1. INTRODUCTION

> “On an uninhabited island off the coast of Maine, tiny wireless sensors deep in the burrows of mysterious sea birds monitor the environmental factors affecting the shy creatures’ comings and goings. In one of Intels chip fabrication facilities, similar sensors measure the subtle vibrations of various machines to detect malfunctions before the equipment breaks down. At an Air Force base across the country, dozens of small sensors scattered across a bogus battlefield outperform tripwires, without the wires. Meanwhile, at the University of California at Berkeley, sensors embedded in a mock buildings walls diagnose its seismic stability after a simulated earthquake. Experimental sensor networks like these are opening up new vistas for scientists and engineers to observe physical phenomena and react to it.” (Source: http://www.intel.com/research/exploratory/instrument_world.htm)

1.1 Overview

Advances in hardware miniaturization and integration have made it possible to design tiny sensor devices that combine sensing with computation, storage and communication capabilities, although limited in capacity. Wireless sensor networks are envisaged to be deployed in a variety of applications including environmental monitoring, wildlife habitat monitoring, wildfire tracking, real-time traffic monitoring, smart homes and many more applications. It is envisioned that sensor networks will continue to penetrate many civil and military applications, which may include hostile environments and sensitive and mission-critical scenarios.

Typically, each sensor collects the data from the physical environment and it needs to send this to a distant home server or a central monitoring location, usually referred to as the base station. The simplest way for the sensors would be to sense and send the data to the base station for analysis.

However, this would generate too much traffic in the network and consequently, aggregation and in-network processing have been proposed as a solution to minimize the traffic in the network. Also, communication over the radio is costlier than local computation which further warrants for opting the aggregation approach.

Further, sensor nodes may be deployed in hostile environments and due to the sheer magnitude of number of nodes deployed in a network, it is infeasible to physically monitor all of them. As such, the network and sensor nodes are susceptible to various types of attacks from adversaries. Particularly, the nodes may be captured or compromised, and all the secret information stored in the nodes would be known to the adversary, who can then easily inject false reports about the phenomenon to be monitored. Such attacks are called false data injection attacks Zhu et al. (2004).

1.2 Motivation and key ideas

The issue of preventing false data injection attacks has attracted substantial research interests Zhu et al. (2004); Ye et al. (2004); Yang and Lu (2004); Yu and Guan (2006); Przydatek et al. (2003). Most existing schemes assume that each individual sensor reports only the sensed reading. So, if the values reported by sensors do not agree to each other, data is considered false and rejected by some process akin to majority voting where all other sensors should agree. Consider a scenario where the phenomenon to be monitored has transient temporal and spatial variations. In this case, different sensors may sense different readings and may not agree to each other all the time. Such transient data, though genuine and important, will be classified by existing schemes as false and rejected. Motivated by this observation, we address the *distinction between genuine transient data vis-a-vis injected false data* in this paper.

Sensor networks are typically organized into clusters. Each cluster has a Cluster Head (CH) responsible for collecting data from sensors in the cluster, doing aggregation and forwarding the result to a distant Base Station (BS). We propose SSTF, a novel **S**ecure **S**tatistical scheme to distinguish data **T**ransience from **F**alse injection in a clustered wireless sensor network. The key ideas of SSTF are twofold. Firstly, each sensor computes a statistical digest of the monitored phenomenon over a moving window of recent readings, and reports this digest along with the current reading to CH. By utilizing the statistical digests to aid in decision making and data aggregation at the CH, SSTF is able to distinguish transient data from false data in most scenarios, which is very difficult if only the current sensed readings are reported by individual sensors. Secondly, SSTF requires the CH to perform a series of carefully-designed inter-sensor statistical tests on both readings and digests reported by individual sensors. Since the false data reported by the compromised node has to pass the inter-sensor tests to escape detection, the impact of false data on the aggregation process is significantly restricted. We enhance the IHHAS (Interleaved Hop-by-Hop Authentication Scheme) scheme proposed in Zhu et al. (2004) and use it as the security framework for SSTF.

1.3 Thesis outline

Following the introduction, this thesis is organized as follows. We discuss the related work in Chapter 2 and give the system model and problem statement in Chapter 3. We describe the statistical framework of the proposed SSTF scheme in Chapter 4, present a realization of SSTF based on the enhanced version of IHHAS in Chapter 5, and analyze its security performance in Chapter 6. Simulation results are presented in Chapter 7. Finally, we present conclusions and future work in Chapter 8.

CHAPTER 2. REVIEW OF LITERATURE

In this chapter, we present some relevant research pertaining to false data filtering, secure aggregation, trust management and outlier detection in wireless sensor networks.

2.1 False Data Filtering

Ye et al. (2004) propose a statistical en-route filtering scheme (SEF), which allows both BS and en-route nodes to detect false data with a certain probability. Zhu et al. (2004) propose an Interleaved Hop-by-Hop Authentication Scheme (IHHAS) where pairwise keys are established between nodes that are $(t+1)$ hops away; with IHHAS, up to t compromised nodes can be tolerated. Yang and Lu (2004) present a commutative cipher based en-route filtering scheme (CCEF) which is based on public-key algorithms that have been reported not suitable for sensor networks due to limited resource capacity of sensor nodes Eschenauer and Gligor (2002). Yu and Guan (2006) present a dynamic en-route false data filtering scheme which alleviates the constraint of fixed path requirement between BS and CH in Zhu et al. (2004); Yang and Lu (2004), thus making the scheme better suited to deal with dynamic topology of sensor networks. Zhang et al. (2006) present an interleaved authentication scheme for filtering false data in multipath routing based sensor networks.

2.2 Secure Aggregation

Przydatek et al. (2003) present SIA for secure aggregation in sensor networks. It provides schemes to compute a few aggregation primitives in a secure manner when queried by trusted outside users. Mahimkar and Rappaport (2004) present SecureDAV which uses Merkle Hash Trees to avoid over-reliance on CH. Since attacker does not know the cluster key, it cannot generate the full signature. Assuming a trusted BS, Wagner (2004) discusses which aggregation functions can be meaningfully computed with resilience. However, Wagner (2004) does not consider in-network aggregation and only BS does the aggregation. Hu and Evans (2003) propose a secure hop-by-hop aggregation scheme that works if a single node is compromised. Yang et al. (2006) present SDAP which organizes sensor nodes into a tree topology and performs a commitment-based hop-by-hop aggregation in each subtree to generate a group-aggregated result.

2.3 Trust Management

Srinivasan et al. (2006) identify four components in a reputation and trust-based system: *information gathering*, *information sharing*, *information modeling*, and *decision making*. Past prominent works dealing with reputation and trust based models for ad-hoc and sensor networks include CORE Michiardi and Molva (2002), CONFIDANT Buchegger and Boudec (2002) and RFSN Ganeriwal and Srivastava (2004). Recently Zhang et al. (2006) present a trust based framework for secure data aggregation referred to as TBFSDA. While CORE and CONFIDANT utilize ratings and reputation tables to implement trust amongst the nodes and are more suitable for mobile ad-hoc networks, RFSN and TBFSDA are designed specifically for wireless sensor networks and they rely on Bayesian and belief propagation models to implement trust and reputation amongst the nodes. In another work, Liu et al. (2007) present a scheme for insider attack detection in wireless sensor networks. Though this work is not explicitly for trust management, one of its contributions is to present a very simple scheme to impart trust values to the nodes. The idea is to compute the standard error of the metric of interest, and only those nodes which lie within twice the standard deviation about the mean can be trusted.

2.4 Outlier Detection

Significant research has been done on outlier detection in very large databases or large scale networks for detecting anomalies using spatial and temporal correlation or learning-based methods Wang, Wang, Hong, and Wan (Wang et al.); Wu and Shao (2005); Oliveira et al. (2006); Chen et al. (2006); Ma and Perkins (2003). These schemes employ complicated techniques such as support vector machines, regression methods or neural networks. As such, due to the complexity of these schemes, they are not applicable to resource-constrained wireless sensor networks. Recently, there has been a lot of focus on outlier detection in sensor networks Subramaniam et al. (2006); Janakiram et al. (2006); Sheng et al. (2007). Janakiram et al. (2006) propose an outlier detection scheme using Bayesian belief networks. However this approach is fairly complex involving a training phase, a testing phase and an inference phase. Subramaniam et al. (2006) present an online outlier detection scheme for sensor data using non-parametric models. The authors propose a framework that computes in a distributed fashion an approximation of multi-dimensional data distributions and demonstrate how the framework can be extended to identify either distance- or density-based outliers in a single pass over the data. Sheng et al. (2007) propose a histogram-based method for outlier detection in sensor networks. In the form of a histogram, hints about the data distribution are collected in a distributed manner. These hints are then utilized to detect outliers for two different definitions of distance-based outliers.

2.5 Novelty of Our Work

There is significant difference between past research and our work. In general, past research has focused on computing various types of single-value aggregates (e.g. sum, count, min, max) securely or accepting the aggregate being correct to a certain probability. Similarly, false data filtering protocols involve accepting or rejecting single values which are proven equal (with some tolerance) or not equal to each other. As discussed earlier, this leads to rejection of even genuine but transient data hence we may not be able to observe the variations in the phenomenon being sensed. Existing trust management and outlier detection schemes primarily focus on the readings reported by individual sensors and/or the spatio-temporal correlation over a history of values amongst different sensors. Since the individual sensors report only their readings, this makes it difficult to judge whether the reading is genuine or false in the case of a time-variant phenomenon. In this thesis, we focus on solving a novel problem: how to observe a time-variant phenomenon by accepting the genuine transient data and at the same time limit the impact of false data injection. In general, it is difficult to distinguish between transient and false data if sensor reading is the only information reported. In our scheme, the sensor nodes report a simple statistical digest along with the reading as opposed to reporting only the reading in existing schemes.

CHAPTER 3. Models and Problem Statement

3.1 System Model

We consider a clustered wireless sensor network that is partitioned into distinct clusters after deployment. Each cluster has a Cluster Head (CH) and a set of sensor nodes, which gather information and report it to CH. CH does decision making and aggregation on the information received from sensors and forwards the result to a distant Base Station (BS). Clustering-related issues such as CH selection and rotation are not the focus of this work.

Distinct clusters could be sensing different phenomena, however we assume that all sensors in a single cluster sense the same phenomenon. The sampling rate of sensors is dependent on the maximum temporal change in the phenomenon as well as the maximum spatial diffusion rate. Instead of sending only the sensed reading to CH, each sensor does a lightweight computation over a moving window of recent sensed readings and sends a simple statistical digest along with the reading to CH periodically.

3.2 Threat Model

Sensor nodes may be compromised or physically captured. All secret information stored in compromised nodes can be accessed by adversaries and they can launch various attacks such as dropping or altering the message contents going through them, so as to prevent BS from receiving authentic sensor readings. Also, there may be colluded attacks where two or more compromised nodes collude to let the false reports escape detection.

3.3 Problem Statement

Due to time and space variant nature of the phenomenon being monitored, instantaneous sensor readings recorded by individual sensors in a cluster may vary. In a monitoring application, it is often critical to observe such transient but genuine data and report them with low false positives. On the other hand, a compromised node (or a group of colluding compromised nodes) will try to inject false reading into the network and our aim is to minimize the impact of false data injection and detect it eventually. Thus, we identify the following design goals for our scheme:

1. it should distinguish genuine transient data from injected false data and report them with low false positives;

2. false data injection should have minimal impact on the aggregation process and be detected as soon as possible; and

3. it should tolerate a large number of compromised nodes.

CHAPTER 4. Proposed SSTF Scheme

SSTF is in short for **S**ecure **S**tatistical scheme to distinguish data **T**ransience from **F**alse injection. It is a two-tier system with a statistical framework on top of a security framework. Such modular design enables us to integrate the statistical framework on top of any existing security scheme with necessary adaptation and enhancement. We present SSTF's statistical framework in this chapter and Chapter 5 describes one particular realization of SSTF by integrating the proposed statistical framework with an enhanced version of the IHHAS security scheme proposed in Zhu et al. (2004).

4.1 Statistical Framework

Statistical framework is the kernel of our proposed SSTF scheme. It consists of four types of operations: *Individual Sensor Behavior*; *Cluster Head Behavior*; *En-route Node Behavior*; and *Base Station Behavior*. Table 4.1 summarizes the notations used in this section.

4.1.1 Individual Sensor Behavior

A sensor node senses the phenomenon at the sampling rate. It maintains a buffer with size equal to the sliding window (w) to store the w most recent readings. Every time a new reading is sensed, the oldest one is deleted; thus a sliding window of size w is implemented at each sensor. We need to have w samples to generate a report. After every reporting interval (n samples), the sensor node v_k computes a simple statistical digest consisting of sample mean (μ_{ki}) and sample variance (σ^2_{ki}) over the sliding window SW_i. This is further illustrated in Fig. 4.1. The report from sensor node v_k to CH is in the format of $R_{ki} \equiv \langle r_{ki}, \mu_{ki}, \sigma^2_{ki} \rangle$.

Table 4.1 Notations Used to Describe the Statistical Framework

Notation	Remarks
$\mathcal{P}$	Phenomenon being monitored by a cluster.
$\mathcal{D}$	Minimum diffusion rate of $\mathcal{P}$, measured in units/sec.
ρ	Phenomenon variation rate: maximum change in the phenomenon per unit time measured in units/sec (e.g. ppm/sec for gas concentration).
x	Sampling rate at each sensor, measured in samples/sec.
d	Maximum distance between any two sensor nodes within a cluster, measured in meters.
n	Reporting interval: each sensor sends a report to CH every n samples.
SW_i	Sliding window for generating the i-th report.
w	Size of the sliding window.
τ	Number of nodes in the cluster (including CH).
v_k	A sensor in the cluster.
u_k	An en-route node.
r_{ki}	Sensed reading reported by v_k in the i-th report.
μ_{ki}	Sample mean reported by v_k in the i-th report.
σ^2_{ki}	Sample variance reported by v_k in the i-th report.
R_{ki}	The i-th report sent by v_k in the format of $\langle r_{ki}, \mu_{ki}, \sigma^2_{ki}\rangle$.
R_{Ag_i}	The i-th aggregated report generated by CH in the format of $\langle r_{Ag_i}, \mu_{Ag_i}, \sigma^2_{Ag_i}\rangle$.

4.1.2 Cluster Head Behavior

In addition to performing the same functions as other sensors in the cluster, CH collects the reports R_{ki} from all individual sensors for testing and aggregation. CH performs two inter-sensor tests. First, CH does a *distribution test* to verify the conformity of the reported digests. Next, by utilizing the reported digests, CH does a *bin test* on the reports that pass the distribution test to limit the impact of false data.

Distribution Test: CH does pairwise tests to check whether the reported distributions $\mathcal{N}(\mu_{ki}, \sigma^2_{ki})$ $(1 \leqslant k \leqslant q)$ conform to each other, where $q \leqslant \tau$ is the number of reporting nodes. A minimum of p nodes need to pass the distribution test for the aggregation to proceed. The number p will be discussed in Chapter 5. CH takes the means reported by sensors as measurements of a common mean. For two sensors v_j and v_k, CH does a *z-test* Hogg (1983) to check whether the means μ_{ji} and μ_{ki} are the same with α confidence level, where α $(0 < \alpha < 1)$ is a design parameter and different α can be achieved by adjusting the sliding window size. The z-test procedure is described in Fig. 4.2.

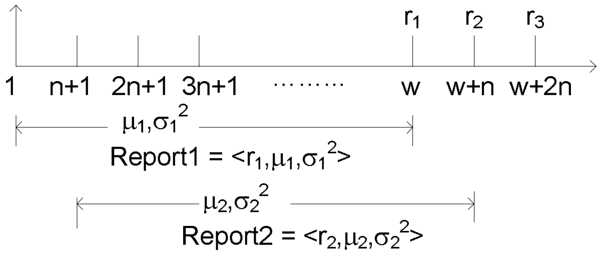


Figure 4.1 Sliding window implementation and report generation at a particular sensor node. Sensor index is omitted. r_i, μ_i, σ_i are respectively the current reading, mean and standard deviation of w samples in SW_i. Shown are reports for two windows (i = 1, 2). There are n non-overlapping samples between two adjacent windows.

z-test for conformity of two sensor reports

For any two sensors v_j, v_k:

$\mathcal{N}(\mu_{ji}, \sigma_{ji}^2)$: Distribution reported by v_j

$\mathcal{N}(\mu_{ki}, \sigma_{ki}^2)$: Distribution reported by v_k

To test whether $\mu_{ji} = \mu_{ki}$:

1. Compute standard deviation of the difference distribution: $\sigma_{jk}^2 = \sigma_{ji}^2 + \sigma_{ki}^2$.
2. Compute standard error of the means: $z_{jk} = \frac{|\mu_{ji} - \mu_{ki}|}{\sqrt{\frac{\sigma_{jk}^2}{w}}}$.
3. The condition for $\mu_{ji} = \mu_{ki}$ with α confidence level is $z_{jk} \leqslant z_\alpha \times \sigma_{jk}$. In this paper we use confidence level of $\alpha = 90\%$; correspondingly $z_\alpha = 0.1257$. Substituting for z_{ij} and σ_{ij}, the condition becomes: $|\mu_{ji} - \mu_{ki}| \leqslant z_\alpha \times \frac{\sigma_{ji}^2 + \sigma_{ki}^2}{\sqrt{w}}$.

Figure 4.2 Application of z-test to digests reported by individual sensors.

If γ ($p \leqslant \gamma \leqslant q$) sensors pass the test, CH proceeds to calculate the aggregated mean and variance based on the sample means and variances reported by the individual sensors that have passed the distribution test. Specifically, CH takes the means reported by individual sensors as measurements of a common aggregated mean that needs to be computed. Under this assumption, the aggregated mean and variance can be computed using Maximum Likelihood Estimation (MLE):

$$\begin{cases} \mu_{Ag_i} = \frac{\sum_{k=1}^{\gamma} \mu_{ki}/\sigma_{ki}^2}{\sum_{k=1}^{\gamma} 1/\sigma_{ki}^2}, \\ \\ \sigma_{Ag_i}^2 = \left(\sum_{k=1}^{\gamma} 1/\sigma_{ki}^2\right)^{-1}. \end{cases} \tag{4.1}$$

Bin Test: We utilize the aggregated variance produced at the end of distribution test to limit the impact of false data on the aggregation process with the following bin test. The bin test is performed only on the readings reported by individual sensors that have passed the distribution test, called the *eligible sensors*. The intuition behind bin test is that, since all sensors observe the same phenomenon which is a diffusion process, the difference between genuine readings reported by any two sensors is most likely to be smaller than twice the aggregated standard deviation (σ_{Ag_i}). Hence, for each eligible sensor v_k, CH utilizes σ_{Ag_i} to form a bin of size $[r_{ki} - 2\sigma_{Ag_i}, r_{ki} + 2\sigma_{Ag_i}]$. Then it checks if the readings reported by other eligible sensors lie in this bin. CH does this for every eligible sensor. Once it knows the bin sizes of all eligible nodes, it picks the one with largest size and averages the readings to get the final aggregated reading r_{Ag_i}. This is illustrated in Fig. 4.3.

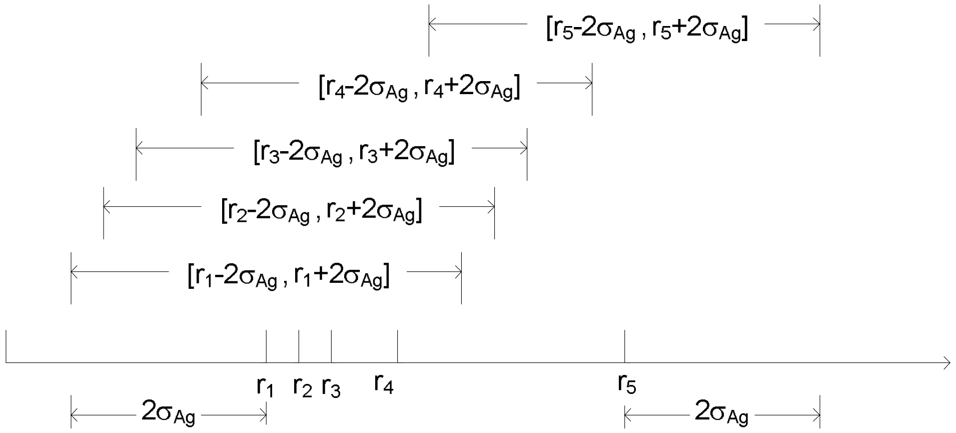


Figure 4.3 Bin Test. Number of sensors in Bin1, Bin2, Bin3, Bin4, Bin5 is 4,4,4,4,1 respectively. Hence we have $r_{Ag} = \frac{1}{4}(r_1 + r_2 + r_3 + r_4)$.

Finally, CH generates the i-th aggregated report $R_{Ag_i} \equiv \langle r_{Ag_i}, \mu_{Ag_i}, \sigma^2_{Ag_i} \rangle$, and merges it with reports received from eligible sensors in the largest bin to form a single message $\mathcal{M}$ and forwards it to BS.

4.1.3 En-route Node Behavior

When an en-route node receives the message, it verifies the integrity of the message. For the first $(t+1)$ en-route nodes from CH, instead of MAC (Message Authentication Code), the actual encrypted data is forwarded and each en-route node tests the conformity between the report by its lower-associated node in the cluster (R_{ki}) and the aggregated report (R_{Ag_i}). Specifically, when an en-route node u_k receives the message generated by CH from its downstream node, it performs the following tests, termed *en-route statistical tests*:

- whether $\sigma_{Ag_i} \leqslant \sigma_{ki}$,
- z-test to check whether $\mu_{Ag_i} = \mu_{ki}$ with α confidence level,
- whether $r_{Ag_i} \in [r_{ki} - 2 \times \sigma_{Ag_i}, r_{ki} + 2 \times \sigma_{Ag_i}]$,

where $R_{ki} = \langle r_{ki}, \mu_{ki}, \sigma_{ki}^2 \rangle$ is the report from u_k's lower-associated node in the cluster. If the tests pass, u_k forwards the message to the next en-route node after suitably modifying the message with proper MAC contents, else it drops the message.

For each en-route node along the remaining path to BS, it verify the integrity of the message by checking the MAC contents. If it is able to verify, it forwards the message to the next en-route node; else it drops the message. More about en-route node filtering will be discussed in conjunction with the security framework in Section 5.3.4.

4.1.4 Base Station Behavior

BS finally verifies the received messages from each CH in the network, and uses them to depict the variations in the monitored phenomena. BS as such, has no major role in the statistical framework.

4.2 Example

Consider a sensor cluster shown in Fig. 4.4. Source, four sensor nodes and CH are randomly placed within a circle with radius of 5 units. The source exhibits random variations in the source data as shown in Fig. 4.5. The window size $w = 100$ samples. Table 4.2 lists the reports generated by the sensors at a particular time instant.

1. Distribution Test: Distribution test is performed on μ_i and σ_i^2 reported by the sensors. For example, for sensors v_1 and v_2, $|\mu_1 - \mu_2| = 1.0759$ which is less than $(z_\alpha \times \frac{\sigma_1^2+\sigma_2^2}{\sqrt{w}} = 18.484)$. It is verified that all the pairwise distribution tests pass. As a result, CH computes the aggregated mean $\mu_{Ag} = 99.9034$ and aggregated variance $\sigma_{Ag}^2 = 146.97$.

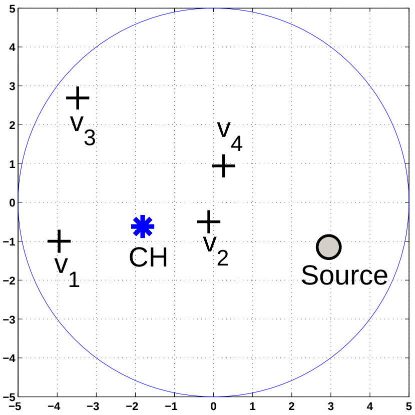


Figure 4.4 An example to illustrate Distribution Test and Bin Test.

Table 4.2 List of reports generated by sensors in the example

Sensor	r	μ	σ^2
v_1	127.719	100.396	735.982
v_2	130.709	99.320	734.46
v_3	127.680	100.572	735.06
v_4	127.761	99.441	734.41
CH	128.519	99.787	734.14

2. Bin Test: The bins are constructed around sensor readings. Since $\sigma_{Ag} = \sqrt{146.97} = 12.1231$, we have

   - Bin1: $\mu_1 \pm 2\sigma_{Ag} \equiv [103.4729, 151.9652]$;
   - Bin2: $\mu_2 \pm 2\sigma_{Ag} \equiv [106.4629, 154.9552]$;
   - Bin3: $\mu_3 \pm 2\sigma_{Ag} \equiv [103.4338, 151.9262]$;
   - Bin4: $\mu_4 \pm 2\sigma_{Ag} \equiv [103.5148, 152.0072]$;
   - BinCH: $\mu_{CH} \pm 2\sigma_{Ag} \equiv [104.2728, 152.7652]$.

   It can be seen that all the sensor readings belong to each of the bins. Thus the largest bin size is 5 and $r_{Ag} = (r_1 + r_2 + r_3 + r_4 + r_5)/5 = 128.4776$.

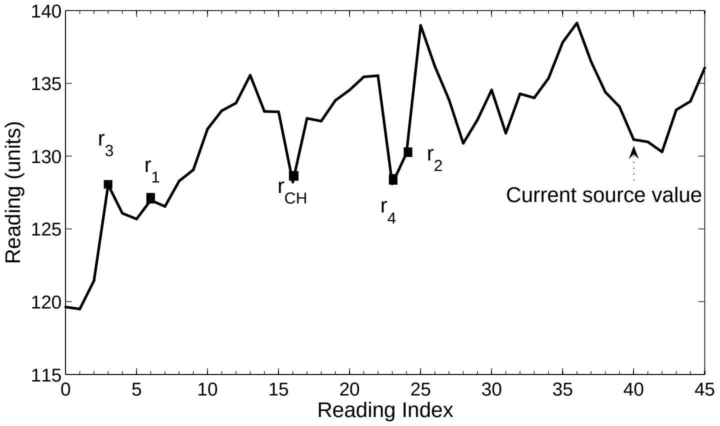


Figure 4.5 Source data. Shown are the readings sensed by the sensors v_i where $i = 1, 2, 3, 4, CH$. There is a delay in the current source value and the reading measured by the sensor depending on the distance between them.

CHAPTER 5. Realization of SSTF with enhanced IHHAS Security Scheme

Here we present a complete realization of SSTF by integrating the afore-described statistical framework with the IHHAS security scheme proposed in Zhu et al. (2004). Since IHHAS is not directly applicable to SSTF, we need to enhance it to meet our requirements. In the rest of this chapter, we first give a brief overview of IHHAS, then describe its limitations and finally present the complete SSTF realization with modified IHHAS.

Similar to IHHAS we make the following security assumptions. Each node shares a master secret key with BS. Each node knows its one-hop neighbors. Pairwise keys can be established between next-hop nodes or nodes that are multiple hops away. All nodes are equally trustable and if a node is compromised, all the information it holds will be exposed. We consider a clustered sensor network and there can be either one-to-one or many-to-one correspondence between the cluster nodes and the en-route nodes to BS. Table 5.1 summarizes the notations used in this section.

5.1 Overview of IHHAS

IHHAS consists of five phases: *node initialization and deployment*, *association discovery*, *report endorsement*, *en-route filtering*, and *base station verification*.

5.1.1 Node Initialization and Deployment

The key server loads each node with a unique ID and necessary keying materials. After deployment, the node establishes a pairwise key with its one-hop neighbors.

Table 5.1 Notations Used to Describe the Security Framework

Notation	Remarks
K_u	Key shared between node u and BS.
K_{uv}	Pairwise key shared between nodes u and v.
F	Family of pseudo-random functions.
K_u^a	Node u's authentication key: $K_u^a = F_{K_u}(0)$.
$u_i \; (1 \leqslant i \leqslant n)$	En-route nodes from CH to BS.
t	Maximum number of tolerable compromised nodes with the original IHHAS.
$v_i \; (1 \leqslant i \leqslant \tau)$	Nodes (including CH) in the cluster ($\tau \geqslant t + 1$).

5.1.2 Association Discovery

This phase is for a node to discover the IDs of its associated nodes. The initial path setup consists of two steps: base station hello and cluster acknowledgment. Incremental association discovery is used to deal with path changes from CH to BS.

5.1.3 Report Endorsement

IHHAS requires that at least $(t + 1)$ nodes agree on the report for it to be considered a valid report. Every participating node computes two MACs (Message Authentication Codes) over the event, one using its shared key with BS (called *individual MAC*) and the other using the shared key with its upper associated node (called *pairwise MAC*). Then it sends the MACs to CH. CH collects MACs from all the participating nodes, authenticates them, wraps them into a single message and forwards to BS. The format of the IHHAS message is as follows (assuming $t = 3$):

$$\begin{aligned} \mathcal{M} \; : \; & E, C_i, \{v_1, v_2, v_3, CH\}, XMAC(E), \\ & \{MAC(K_{CHu_4}, E), MAC(K_{v_3u_3}, E), MAC(K_{v_2u_2}, E), MAC(K_{v_1u_1}, E)\}, \end{aligned} \tag{5.1}$$

where $MAC(K_{v_iu_i}, E), i = 1, 2, 3, 4$ are the pairwise MACs and XMAC is a compressed MAC computed by CH using individual MACs as given below:

$$XMAC(E) = MAC(K_{v1}^a, E) \oplus MAC(K_{v2}^a, E) \oplus MAC(K_{v3}^a, E) \oplus MAC(K_{v4}^a, E). \tag{5.2}$$

5.1.4 En-route Filtering

Each en-route node verifies the MAC computed by its lower associated node, and then removes the MAC from the received message. If verification succeeds, it attaches a new MAC to the message based on the pairwise key with its upper associated node and forwards it to BS.

5.1.5 Base Station Verification

BS verifies the report after receiving the message. If the BS detects that at least $(t+1)$ nodes have endorsed the report correctly, it accepts the report; otherwise, it discards the report.

5.2 Limitations of IHHAS

While IHHAS works well with the system model described in Zhu et al. (2004), it overlooks the following scenarios.

5.2.1 Large cluster size not addressed

Designed with the implicit assumption that there are exactly $(t+1)$ nodes in a cluster (including CH), IHHAS works well as long as the number of compromised nodes (within cluster or en-route) is no larger than t. With more than $(t+1)$ nodes in the cluster, the association discovery phase of IHHAS works incorrectly since it can not guarantee a unique lower-associated node to an en-route node. In this paper, we generalize IHHAS to accommodate more than $(t+1)$ cluster nodes.

5.2.2 ID attack not considered

The format of the IHHAS message is given in Eq. (5.1). All en-route nodes only check the pairwise MACs but do not verify the IDs of sensor nodes endorsing the reports. Only BS can verify the node IDs and the XMAC. This makes the scheme prone to *ID attack*, where adversary can simply modify the node ID list $\{v_1, v_2, v_3, CH\}$ so that BS cannot verify the XMAC with the modified list hence rejects the report; in this case, all en-route nodes waste energy in forwarding the message. To overcome this limitation, we make a simple improvement to IHHAS wherein each node includes its node ID in the

MAC contents and the en-route nodes verify node ID in the list with node ID carried in the MAC contents.

5.2.3 Not suitable for distinct reports from sensors

IHHAS works perfectly when all sensors agree on an event E, which means that E could be a logical or boolean value so that all sensors agree on exactly the same thing. For example, sensors responding either “Yes” or “No” to a query whether the room temperature is higher than 150^oF, would be such an event. In a scenario where CH needs to do aggregation and sensors could report possibly different readings and digests which would generally be the case in practice, computation of XMAC as in Eq. (5.2) is not possible. Further, it may incur significant communication overheads to forward all individual MACs to BS instead of compressing them.

5.3 Enhancing IHHAS to Integrate with SSTF

To address above inadequacies, following modifications are done to the association discovery phase; reporting by individual sensors; testing, aggregation and final message preparation by CH; and the en-route filtering phase in IHHAS.

5.3.1 Association Discovery

There are totally $\tau \geqslant t+1$ nodes (including CH) in the cluster. As in IHHAS, BS sends *Hello* message to enable a node to discover its upper associated node. On receiving a Hello message from BS, a node attaches its own ID to the Hello message before re-broadcasting it. The maximum number of node IDs that are included in the Hello message is $t+1$. CH divides the cluster nodes (including itself) into $(t+1)$ groups, g_i ($1 \leqslant i \leqslant t+1$), and each group has a minimum of one and a maximum of $\psi = \lceil \frac{\tau}{t+1} \rceil$ nodes. When CH receives the Hello message containing $(t+1)$ IDs from its previous hop node, it assigns the $(t+1)$ IDs to the $(t+1)$ groups. Thus, all the nodes in a group have a single upper associated node. Also, CH keeps a list of the nodes in each group.

Example: Fig. 5.1 illustrates the association discovery process (BS “*Hello*” and cluster “*ACK*”). There are a total of $\tau = 10$ nodes. When CH receives the Hello message $(u4, u3, u2, u1)$, it divides

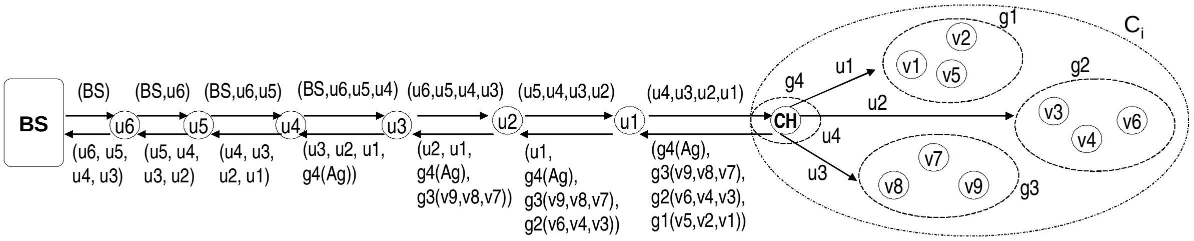


Figure 5.1 An example to show the association discovery process for $t+1=4$ and $\tau=10$. The cluster head CH divides the cluster nodes (including itself) into $t+1$ groups and assigns a group ID to each group. Each group has a maximum of $\psi = \lceil \tau/(t+1) \rceil = 3$ nodes. BS is the base station. The contents of "*Hello*" and "*ACK*" messages are shown inside the parenthesis $(\cdot)$.

all the nodes into four groups: $(g1(v5, v2, v1),\ g2(v6, v4, v3),\ g3(v9, v8, v7),\ g4(CH))$. There can be a maximum of three nodes in a group. Then it assigns each of $(u4, u3, u2, u1)$ to $(g4, g3, g2, g1)$ respectively. During the cluster ACK process, the ACK messages carry group ID along with the lists of nodes for each group so that en-route nodes get to know its lower associated nodes. For example, when $u1$ receives $(g4(CH),\ g3(v9, v8, v7),\ g2(v6, v4, v3),\ g1(v5, v2, v1))$, it knows that its lower associated nodes are $v5, v2, v1$ in group $g1$. It then removes this group and substitutes its ID $u1$ in the beginning of the message and forwards $(u1, g4(CH), g3(v9, v8, v7),\ g2(v6, v4, v3))$ to $u2$. The process proceeds similarly at each en-route node. ■

We define *Report Limit* (θ) as the maximum number of reports from a group that can be used by CH. It is easy to see that $\theta \leqslant \psi$. CH needs at least $(t+1)$ reports, and at most θ reports from each group will be used. To satisfy this, our scheme needs reports from at least p nodes in the cluster, where p is given by:

$$p \ = \ max\left(t+1, \psi \times \left(\left\lceil \frac{(t+1)}{\theta} \right\rceil - 1\right) + \theta\right). \tag{5.3}$$

Note that the maximum number of compromised nodes our scheme can tolerate is still t; however the en-route filtering phase will work only if less than $(t+1)/\theta$ en-route nodes are compromised. If $N_c\ \left(\frac{t+1}{\theta} \leqslant N_c \leqslant \tau\right)$ nodes are compromised, though BS will eventually detect the false report, the en-route nodes may not be able to detect it and may keep on forwarding the message. θ is also useful in the en-route filtering phase which will be discussed in Section 5.3.4.

5.3.2 Reporting by Individual Sensors

Each sensor node v_k in the cluster generates the statistical digest $R_{ki} \equiv \langle r_{ki}, \mu_{ki}, \sigma^2_{ki} \rangle$ and signs it with the key shared with its upper associated en-route node, $K_{v_k u_k}$. Then v_k sends $(R_{ki}, E_{K_{v_k u_k}}(R_{ki}, v_k))$ to CH. Since CH doesn't have knowledge of $K_{v_k u_k}$, it can't modify the second encrypted term in the above tuple received from v_k.

5.3.3 Testing, Aggregation and Final Message Preparation

CH receives reports from nodes in its cluster and performs distribution test and bin test on the received reports as detailed in Section 4.1. Then CH picks $(t+1)$ nodes out of the eligible sensors belonging to the largest bin with the following rules: choose from as many groups as possible, and select at most θ nodes from each group.

The message $\mathcal{M}$ that CH finally generates and forwards to BS consists of the aggregated report R_{Ag_i}, cluster ID C_i, ID list of the selected $(t+1)$ nodes and $(t+1)$ distinct encrypted reports $E_{K_{v_k u_k}}(R_{ki}, v_k)$ for each chosen v_k, and a special counter κ initially set to zero. For example, in Fig. 5.1, suppose v_1, v_2, v_3, v_4 are chosen from the largest bin, then the message $\mathcal{M}$ generated by CH is:

$$\begin{aligned}\mathcal{M} \equiv\ & \langle R_{Ag_i}, C_i, \kappa = 0, \{v_1, v_2, v_3, v_4\}, \{E_{K_{v_4 u_2}}(R_{4i}, v_4), E_{K_{v_3 u_2}}(R_{3i}, v_3), \\ & E_{K_{v_2 u_1}}(R_{2i}, v_2), E_{K_{v_1 u_1}}(R_{1i}, v_1)\}\rangle.\end{aligned} \tag{5.4}$$

The order of the encrypted reports in $\mathcal{M}$ corresponds to that in the cluster ACK message during the association discovery phase so that a node receiving $\mathcal{M}$ knows which reports could be from its lower-associated nodes. κ is a special counter updated by en-route nodes to keep in track how many consecutive nodes have not been able to verify any of the reports. It will be described more in the en-route filtering phase presented next.

5.3.4 En-route Filtering

Notice that the message $\mathcal{M}$ sent by CH consists of encrypted reports. We can reduce the size of this message by using MACs instead of encrypted reports. Unfortunately, this is not possible for the first $(t+1)$ en-route nodes. If u_k is one of the first $(t+1)$ en-route nodes, when it receives $\mathcal{M}$, it performs en-route statistical tests to check the conformity of the report R_{ki} by its lower associated node v_k in the cluster with the aggregated report R_{Ag_i}. If the tests pass, u_k replaces the encrypted report with a pairwise MAC which consists of R_{Ag_i} and the node ID v_k and is signed using the pairwise key shared with its upper-associated en-route node. Thus after $(t+1)$ hops, all the encrypted reports are replaced by smaller size MACs.

For the remaining path to BS, when an en-route node u_k receives $\mathcal{M}$ from its downstream node, it checks whether the number of different pairwise MACs in $\mathcal{M}$ is $t+1$. u_k tries to verify the last θ MACs in the pairwise MAC list, based on the pairwise key(s) shared with its lower-associated node(s). If u_k is unable to verify any of the pairwise MACs, it increments κ by 1 and forwards the message to its upstream node.

If at any point of time $\kappa \geqslant \lceil (t+1)/\theta \rceil$, it implies that more than $\lceil (t+1)/\theta \rceil$ consecutive en-route nodes have been compromised and the message will be dropped. On the other hand, en-route filtering phase of the scheme will not work if more than $\lceil (t+1)/\theta \rceil$ nodes are compromised, since the compromised nodes may reset κ to zero.

If u_k is able to verify ν ($\leqslant \theta$) nodes and if u_k is more than $(t+1)$ hops away from BS, it proceeds to compute ν new pairwise MACs over the report R_{ki} ($1 \leqslant k \leqslant \nu$) using the pairwise key shared with its upper-associated node. It then removes the last ν MACs from the MAC list and inserts the ν new MACs at the beginning of the MAC list. Finally it resets κ to zero and forwards the message to its upstream node.

Example: Consider the first en-route node u_1 in Fig. 5.2. When node u_1 receives the message $\mathcal{M}$

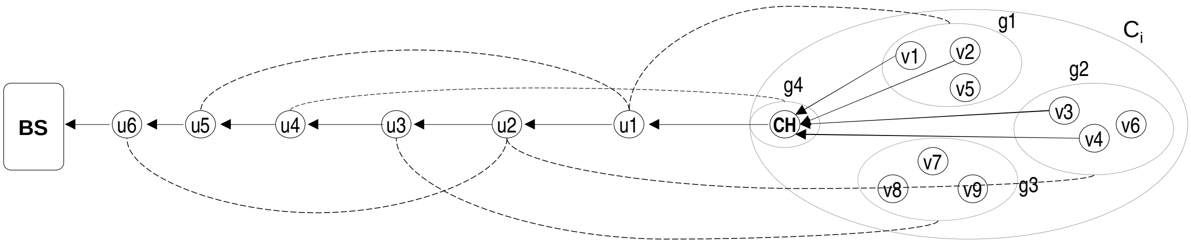


Figure 5.2 An example to show the en-route filtering process. Reports from nodes v_1, v_2, v_3, v_4 belong to the largest bin and are chosen by CH to be a part of the final message. Node associations are shown as dashed lines.

given by Eq. (5.4) from CH, it checks whether there are four encrypted reports. u_1 first tries to decrypt the last report using $K_{u_1v_1}$. Since it is able to decrypt, it then verifies the conformity of R_{Ag_i} and R_{1i}. If tests pass, u_1 replaces $E_{K_{v_1u_1}}(R_{1i}, v_1)$ with $MAC(K_{u_1u_5}, R_{Ag_i}, v_1)$. u_1 then tries to decrypt the next report using $K_{u_1v_2}$ which also holds true. Then u_1 does tests for conformity of R_{Ag_i} and R_{2i}. If tests pass, u_1 replaces $E_{K_{v_2u_1}}(R_{2i}, v_2)$ with $MAC(K_{u_1u_5}, R_{Ag_i}, v_2)$. At this point, u_1 has verified $\theta = 2$ encrypted reports and it would not attempt verifying any more reports. Since u_1 was able to verify the MACs, it resets the counter $\kappa = 0$ and forwards the following message to u_2:

$$\begin{aligned} \mathcal{M} \ \equiv \ & \langle R_{Ag_i}, C_i, \kappa = 0, \{v_1, v_2, v_3, v_4\}, \{MAC(K_{u_1u_5}, R_{Ag_i}, v_2), \\ & MAC(K_{u_1u_5}, R_{Ag_i}, v_1), E_{K_{v_4u_2}}(R_{4i}, v_4), E_{K_{v_3u_2}}(R_{3i}, v_3)\}\rangle. \end{aligned} \tag{5.5}$$

Similarly, when u_2 receives R, it is also able to verify the last two encrypted reports and replaces them with MACs signed with the pairwise key shared with its upper-associated node u_6. The message that u_2 forwards to u_3 becomes:

$$\begin{aligned} \mathcal{M} \ \equiv \ & \langle R_{Ag_i}, C_i, \kappa = 0, \{v_1, v_2, v_3, v_4\}, \{MAC(K_{u_2u_6}, R_{Ag_i}, v_4), \\ & MAC(K_{u_2u_6}, R_{Ag_i}, v_3), MAC(K_{u_1u_5}, R_{Ag_i}, v_2), MAC(K_{u_1u_5}, R_{Ag_i}, v_1)\}\rangle. \end{aligned} \tag{5.6}$$

Please note that at this point all the encrypted reports have been replaced with MACs and henceforth the message being forwarded has lower communication costs. Please also note that the MACs are signed over the aggregated report R_{Ag_i}. ■

CHAPTER 6. Security Analysis

Individual sensor nodes or CH can lie about the measurements, digests and aggregated reports. All these attacks are collectively referred to as *content attacks*. In this section, we present detailed analysis on various content attacks. Throughout the analysis, for the sake of simplicity, the index of sliding window in the reports is omitted.

6.1 Content Attacks by Individual Sensors

6.1.1 Effect of False Injection

A pairwise distribution test is performed to test the equality of means reported by the sensor nodes. Let v_j be a compromised node with true mean and variance of (μ, σ^2). Assume v_j reports (μ', σ'^2) instead of the true values. To pass the distribution tests, the following conditions should hold:

$$|\mu' - \mu_k| \leqslant z_\alpha \frac{\sigma'^2 + \sigma_k^2}{\sqrt{w}}; \quad \forall k \in [1, \tau], k \neq j, \tag{6.1}$$

where μ_k and σ_k^2 are respectively the mean and variance reported by sensor v_k, τ is the number of nodes in the cluster, and w is size of the sliding window.

The reading reported by the sensor should be within certain limits to pass the bin test. The compromised node wants a false reading $r' = r + \Delta_r$ to get accepted, where r is the true reading measured by the sensor. We are interested in computing the maximum possible expected distortion that an attacker can inject without being detected i.e. we want to maximize $E[\Delta_r | \Delta_r \text{ is accepted}]$. Consider Fig. 6.1. r is a valid measurement. Let σ'^2_{Ag} denote the false aggregated variance when the compromised node reports a false variance σ'^2. Then $2\sigma'_{Ag}$ would be the maximum possible difference between the readings allowed by the bin test. r_{min} and r_{max} are respectively the true minimum and maximum reading amongst all the sensors. Assume that the readings follow a uniform distribution over $[r_{min}, r_{max}]$. Let

$\mathcal{W}_r = r_{max} - r_{min}$ denote the width of this interval. Then the maximum possible reading that can escape the bin test is $r_{min} + 2\sigma'_{Ag}$. For a given r, let $I_A(\Delta_r)$ denote the indicator function whether

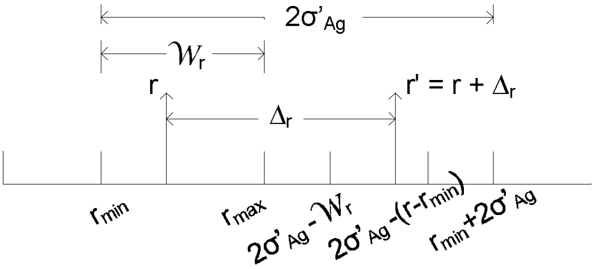


Figure 6.1 Imposing limits based on the Bin Test.

Δ is accepted, i.e.

$$I_A(\Delta_r) = \begin{cases} 1, & 0 \leqslant \Delta_r \leqslant r_{min} + 2\sigma'_{Ag} - r, \\ 0, & \text{Otherwise.} \end{cases}$$

Then we have:

$$\begin{aligned} E[\Delta_r|\Delta_r \text{ is accepted}] &= \int_{r_{min}}^{r_{max}} I_A(\Delta_r)\Delta_r \frac{dr}{r_{max} - r_{min}} \\ &= \begin{cases} \Delta_r, & \Delta_r \leqslant 2\sigma'_{Ag} - \mathcal{W}_r, \\ \frac{(2\sigma'_{Ag} - \Delta_r)\Delta}{\mathcal{W}_r}, & 2\sigma'_{Ag} - \mathcal{W}_r < \Delta_r \leqslant 2\sigma'_{Ag}, \\ 0, & \Delta > 2\sigma'_{Ag}. \end{cases} \end{aligned} \tag{6.2}$$

Differentiating $E[\Delta_r]$ with respect to Δ_r, we get the optimal $\Delta_r = \Delta_r^*$ given by:

$$\Delta_r^* = \begin{cases} 2\sigma'_{Ag} - \mathcal{W}_r, & \mathcal{W}_r < \sigma'_{Ag}, \\ \sigma'_{Ag}, & \mathcal{W}_r \geqslant \sigma'_{Ag}. \end{cases} \tag{6.3}$$

Subsequently, the maximum expectation is given by:

$$E_{max} = \begin{cases} 2\sigma'_{Ag} - \mathcal{W}_r, & \mathcal{W}_r < \sigma'_{Ag}, \\ \frac{\sigma'^2_{Ag}}{\mathcal{W}_r}, & \mathcal{W}_r \geqslant \sigma'_{Ag}. \end{cases} \tag{6.4}$$

Fig. 6.2 illustrates the variation of expectation with respect to Δ_r. We can see that Δ_r is dependent on the aggregated variance σ'^2_{Ag} and $\mathcal{W}_r$. When the source variation is less, $\mathcal{W}_r$ is small and the compromised node should report $r' = r + 2\sigma'_{Ag} - \mathcal{W}_r$; in case of a highly varying source, $\mathcal{W}_r$ is

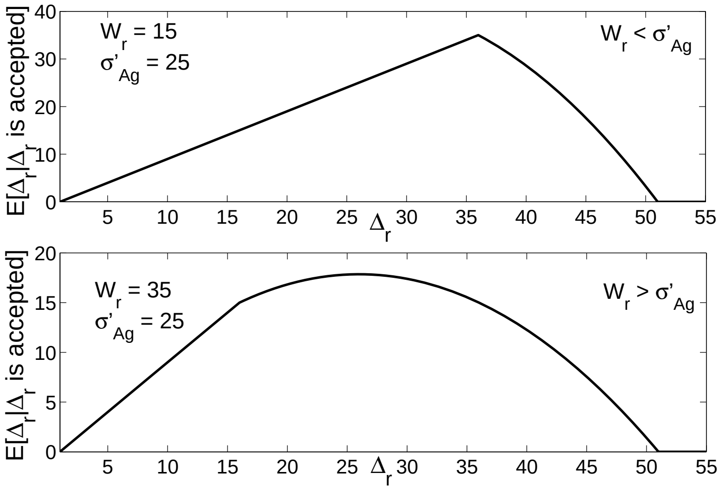


Figure 6.2 E[$\Delta_r|\Delta_r$ is accepted] vs. Δ_r.

large and the compromised node should report $r' = r + \sigma'_{Ag}$. The impact of the false injection on the aggregated reading (r_{Ag}), denoted by $\mathcal{F}$, is:

$$\mathcal{F} = \frac{E_{max}}{\text{Number of Readings in the Largest Bin}}, \tag{6.5}$$

where E_{max} is given by Eq. (6.4).

6.1.2 Attack Strategies

There are two strategies for an adversary to inject false data and distort r_{Ag}. As can be seen from Eq. (6.5), the adversary can either attempt to maximize Δ_r or minimize the number of readings in the largest bin to increase $\mathcal{F}$.

Strategy 1: The first strategy is to report a small false variance such that the aggregated variance and hence the bin width is reduced. This is equivalent to decreasing the denominator in Eq. (6.5). As a result some of the genuine readings are excluded from aggregation and hence, the false data injected by the adversary can have more impact. However, due to reduced bin width, the distortion Δ_r that can be introduced into the reading is also small.

Strategy 2: The other strategy is to report a large variance such that the aggregated variance is increased. This results in a larger bin width and hence, larger distortion Δ_r can be introduced into the reading.

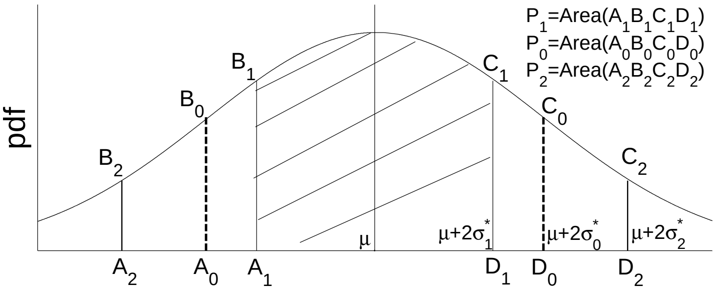


Figure 6.3 Demonstrating the relation between the bin size and the aggregated standard deviation.

6.1.3 Selecting the Best Strategy

Let $\sigma_{Ag}^{(0)}$ be the genuine aggregated standard deviation and $\sigma_{Ag}^{(1)}$ and $\sigma_{Ag}^{(2)}$ be the false aggregated standard deviation computed using Strategy 1 and Strategy 2 respectively. Consider Fig. 6.3. Let P_0, P_1, P_2 denote the probability that a reading lies in the bins $[A_i, D_i]$ $(i.e., \mu \pm 2\sigma_{Ag}^{(i)}; i = 0, 1, 2)$ for the cases of no compromised node, one compromised node using Strategy 1, and one compromised node using Strategy 2, respectively. $\mathcal{E}$ is the number of eligible sensors. The bin size i.e. the expected number of readings that lie in the bins $[A_0, D_0]$, $[A_1, D_1]$, $[A_2, D_2]$ is given by $\mathcal{E}P_0$, $\mathcal{E}P_1$, $\mathcal{E}P_2$ respectively. From Fig. 6.3, we can see that:

$$\frac{P_1}{P_2} = \frac{Area(A_1B_1C_1D_1)}{Area(A_2B_2C_2D_2)} > \frac{\sigma_{Ag}^{(1)}}{\sigma_{Ag}^{(2)}} \implies \frac{\sigma_{Ag}^{(1)}}{P_1} < \frac{\sigma_{Ag}^{(2)}}{P_2}. \tag{6.6}$$

The adversary introduces a distortion of Δ_r^* according to Eq. (6.3). The attack can be analyzed for the following three cases.

- Case 1: $W_r < \sigma_{Ag}^{(1)} < \sigma_{Ag}^{(2)}$:

  Using Eqs. (6.4) and (6.5), the effect on false reading using Strategy 1 and Strategy 2 is $\mathcal{F}_1 = \frac{2\sigma_{Ag}^{(1)} - \mathcal{W}_r}{\mathcal{E}P_1}$ and $\mathcal{F}_2 = \frac{2\sigma_{Ag}^{(2)} - \mathcal{W}_r}{\mathcal{E}P_2}$ respectively. From Eq. (6.6), it can be easily seen that:

  $$\frac{2\sigma_{Ag}^{(1)} - \mathcal{W}_r}{\mathcal{E}P_1} < \frac{2\sigma_{Ag}^{(2)} - \mathcal{W}_r}{\mathcal{E}P_2}, \tag{6.7}$$

  implying $\mathcal{F}_2 > \mathcal{F}_1$.

- Case 2: $\sigma_{Ag}^{(1)} < W_r < \sigma_{Ag}^{(2)}$:

  In this case, the effect on false reading using Strategy 1 and Strategy 2 is $\mathcal{F}_1 = \frac{\sigma_{Ag}^{(1)2}}{\mathcal{E}P_1\mathcal{W}_r}$ and

$\mathcal{F}_2 = \frac{2\sigma_{Ag}^{(2)} - \mathcal{W}_r}{\mathcal{E}P_2}$ respectively. From Eq. (6.6), we have:

$$\begin{aligned}\frac{\sigma_{Ag}^{(1)}}{P_1} < \frac{\sigma_{Ag}^{(2)}}{P_2} \Longrightarrow \frac{\sigma_{Ag}^{(1)}}{\mathcal{E}P_1} &< \frac{\sigma_{Ag}^{(2)} + (\sigma_{Ag}^{(2)} - \mathcal{W}_r)}{\mathcal{E}P_2} \quad (\because \sigma_{Ag}^{(2)} > \mathcal{W}_r)\\ \Longrightarrow \frac{\sigma_{Ag}^{(1)}}{\mathcal{E}P_1} &< \frac{2\sigma_{Ag}^{(2)} - \mathcal{W}_r}{\mathcal{E}P_2}.\end{aligned} \tag{6.8}$$

Since $\sigma_{Ag}^{(1)} < W_r$, multiplying the LHS of above equation by $\frac{\sigma_{Ag}^{(1)}}{W_r}$ and RHS by 1, we get

$$\frac{\sigma_{Ag}^{(1)}}{\mathcal{E}P_1} \cdot \frac{\sigma_{Ag}^{(1)}}{W_r} < \frac{2\sigma_{Ag}^{(2)} - \mathcal{W}_r}{\mathcal{E}P_2} \cdot 1 \Longrightarrow \frac{\sigma_{Ag}^{(1)2}}{\mathcal{E}P_1\mathcal{W}_r} < \frac{2\sigma_{Ag}^{(2)} - \mathcal{W}_r}{\mathcal{E}P_2}, \tag{6.9}$$

implying $\mathcal{F}_2 > \mathcal{F}_1$.

- Case 3: $\sigma_{Ag}^{(1)} < \sigma_{Ag}^{(2)} < W_r$:

  In this case, the effect on false reading using Strategy 1 and Strategy 2 is $\mathcal{F}_1 = \frac{\sigma_{Ag}^{(1)2}}{\mathcal{E}P_1\mathcal{W}_r}$ and $\mathcal{F}_2 = \frac{\sigma_{Ag}^{(2)2}}{\mathcal{E}P_2\mathcal{W}_r}$ respectively. From Eq. (6.6), it can be easily seen that

$$\frac{\sigma_{Ag}^{(1)2}}{\mathcal{E}P_1\mathcal{W}_r} < \frac{\sigma_{Ag}^{(2)2}}{\mathcal{E}P_2\mathcal{W}_r}, \tag{6.10}$$

  implying $\mathcal{F}_2 > \mathcal{F}_1$.

From the three exhaustive cases discussed above, we conclude that the attacker can cause maximum distortion in r_{Ag} when it adopts Strategy 2. So, the compromised node should report a high variance. For maximum impact, the adversary reports a fake variance equal to ∞. Hence, the adversary reports $\sigma'^2 = \infty$. Consequently, regardless of the false μ' being reported, the resultant $\mu_{Ag}^{(2)*}$ and $\sigma_{Ag}^{(2)*2}$ is:

$$\begin{cases} \mu_{Ag}^{(2)*} &= \frac{\sum_{\substack{k=1\\k\neq j}}^{\gamma} \mu_k/\sigma_k^2}{\sum_{\substack{k=1\\k\neq j}}^{n} 1/\sigma_k^2}, \\ \sigma_{Ag}^{(2)*2} &= \left(\sum_{\substack{k=1\\k\neq j}}^{\gamma} 1/\sigma_k^2\right)^{-1}, \end{cases} \tag{6.11}$$

where j is the index of the compromised sensor node.

Further, from Eq. (6.5), the effect on r_{Ag} is:

$$\mathcal{F} = \begin{cases} \frac{2\sigma_{Ag}^{(2)*} - \mathcal{W}_r}{\mathcal{E}P_2}, & \mathcal{W}_r < \sigma_{Ag}^{(2)*}, \\ \\ \frac{\sigma_{Ag}^{(2)*2}}{\mathcal{E}P_2\mathcal{W}_r}, & \mathcal{W}_r \geqslant \sigma_{Ag}^{(2)*}, \end{cases} \tag{6.12}$$

where $\sigma_{Ag}^{(2)*}$ is given by Eq. (6.11).

6.2 Content Attacks by the Cluster Head

CH produces an aggregated report R_{Ag} which is verified by the en-route nodes for conformity with the individual sensor reports R_k. The worst case performance of the system occurs when CH is compromised. This happens because compromised CH can lie about the aggregated report $R_{Ag} = \langle r_{Ag}, \mu_{Ag}, \sigma^2_{Ag} \rangle$. The following conditions should hold for R_{Ag} to pass the en-route statistical tests:

- From Eq. (4.1), we can see that $\sigma_{Ag} \leqslant min(\sigma_k)$ for each sensor v_k whose report is included in the final message $\mathcal{M}$. Hence R_{Ag} with a larger σ_{Ag} will be rejected. To alter μ_{Ag} and r_{Ag}, CH chooses the largest possible σ_{Ag} given by: $\sigma'_{Ag} = min(\sigma_k)$.

- Each of the first $(t+1)$ en-route nodes, say u, performs distribution test to test the equality of the aggregated mean μ_{Ag} with the mean μ_k reported by its lower associated node v_k in the cluster. The maximum false μ'_{Ag} that would satisfy the distribution test is given by:

$$\mu'_{Ag} = min\left(\mu_k + z_\alpha \frac{min(\sigma^2_k) + \sigma^2_k}{\sqrt{w}}\right), \qquad (6.13)$$

  where k is index of the eligible sensors (from the largest bin) whose reports are included in the final message.

- Further, the aggregated reading r_{Ag} should satisfy bin test at each of the first $(t+1)$ en-route nodes. Let r_{Ag} be the true aggregated reading, and r'_{Ag} be the maximum acceptable false reading reported by compromised CH. It is easy to see that, if CH reports $r'_{Ag} = min(r_k) + 2min(\sigma_k)$, it will always be accepted. Thus, CH can distort the true readings by a maximum of $r_{Ag} - min(r_k) + 2min(\sigma_k)$.

Since our security framework is based on IHHAS, our scheme is equally resilient as IHHAS to other security attacks, such as outsider attacks, replay attacks, cluster insider attacks and en-route insider attacks. Discussions on those security attacks are omitted since they are not the focus of our work.

6.3 Summary

We mentioned in Section 5.3.1, the maximum number of compromised nodes our system can accept is t, where t is a system parameter. ($t + 1$ is the number of encrypted reports / MACs kept in the messages forwarded to the base station). However, this statement is for the general case, where both the cluster nodes and cluster head are compromised. We can achieve a much better performance if the cluster head is not compromised. This is due to the fact that if a large number of non-compromised nodes are present in the cluster, the bin test will be successful to filter out the non-conforming false data injected by the adversaries. False data conforming to the bin test, on the other hand, will not significantly change the aggregated result. Hence when CH is not compromised, the system can tolerate $\max(t, \text{majority - 1})$ number of compromised nodes. For example, if t is 3, number of cluster nodes is 12, the system can accept maximum 5 compromised nodes; if t is 5, number of cluster nodes is 8, the system can accept maximum 5 compromised nodes.

CHAPTER 7. Performance Evaluation

We study the performance of SSTF by simulation. We compare SSTF with a simple majority voting scheme to show its effectiveness in preserving transient data. We also demonstrate the limited impact of false data with SSTF in the presence of compromised nodes under various attack strategies, and compare it with two recent works, one being an outlier detection scheme and the other a reputation-based scheme.

7.1 Simulation Setup

The wireless sensor network is divided into circular clusters. Each cluster is responsible for sensing the time-varying phenomenon in its region. We focus on one particular cluster shown in Fig. 7.1 to demonstrate our scheme. Cluster nodes are randomly placed in the circular region and one of the nodes is CH. A single source is present at a random location in the cluster. The phenomenon exhibits a radial diffusion pattern, implying that the sensors nearest to the source sense the change first. Table 7.1 lists the parameters used for simulation.

Table 7.1 Simulation Parameters

Parameter	Notation	Value
Phenomenon variation rate	ρ	10 units/sec
Maximum inter-sensor distance	d	10 meters
Diffusion rate	$\mathcal{D}$	2 units/sec
Sampling rate	x	10 samples/sec
Reporting interval	n	25 samples
Sliding window size	w	1000 samples
Number of nodes in the cluster	τ	10
Random measurement error at sensors		$\mathcal{N}(0, 0.01)$

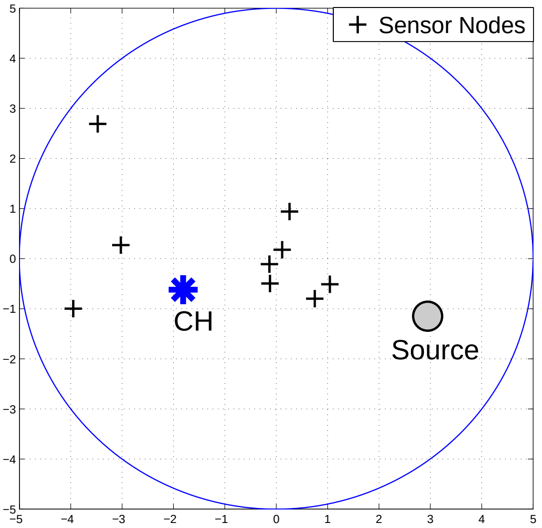


Figure 7.1 Simulation setup.

7.2 Simulation Results

We conduct various simulations to demonstrate the effectiveness of SSTF in meeting its design goals viz. preservation of data transience and limiting the impact of false data injection.

7.2.1 Preservation of Data Transience

We consider the performance of our scheme in the presence of no compromised nodes. The phenomenon varies from 0 units/second to 50 units/second which amounts to a change of 0 units/sample to 5 units/sample. Fig. 7.2 plots the simulation results. In our scheme most of the times all the genuine data is preserved regardless of transient variations. It is observed that when the variation rate is small, up to 30% of the nodes are excluded from participating in the aggregation. This happens because the bin size becomes very small when the source is constant. However, this doesn't hamper the ability of our scheme to monitor transient data since some genuine nodes are excluded from the largest bin only when source data is itself constant and there is negligible impact on r_{Ag}.

We compare our scheme to a simple majority voting scheme where the nodes agree if the readings reported are within random measurement error of each other. When there is no variation, the readings are pretty constant and all the nodes agree. However, as the variation rate increases, the readings amongst sensors do not agree with each other anymore, and more and more genuine data are excluded

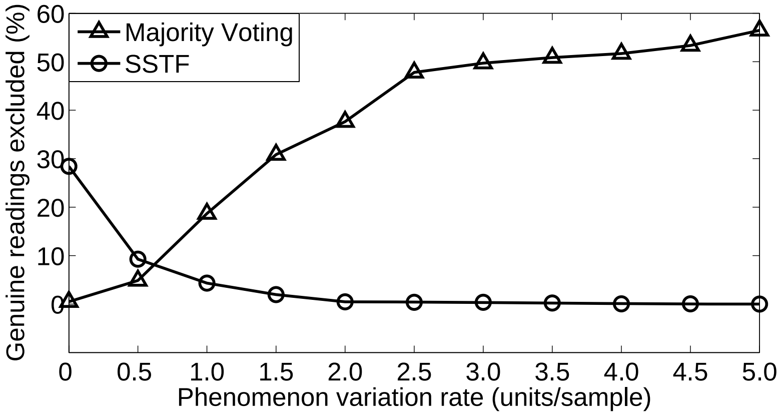


Figure 7.2 Percentage of genuine readings excluded vs. Phenomenon variation rate.

from aggregation. Thus the system starts losing "important" information during data transience which is not desired. We can see in Fig. 7.2, almost 60% genuine data is lost at high variation rate.

7.2.2 Limiting the Impact of False Data Injection

In Fig. 7.3, we show the effect of false data with respect to different phenomenon variation rates and false injection.

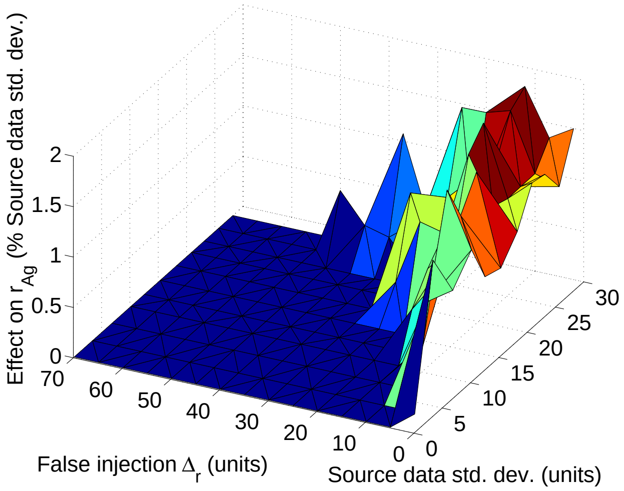


Figure 7.3 False effect vs. Phenomenon variation and false injection. Phenomenon variation is indicated by source data standard deviation and z-axis shows the impact on r_{Ag} as a percentage of source data standard deviation.

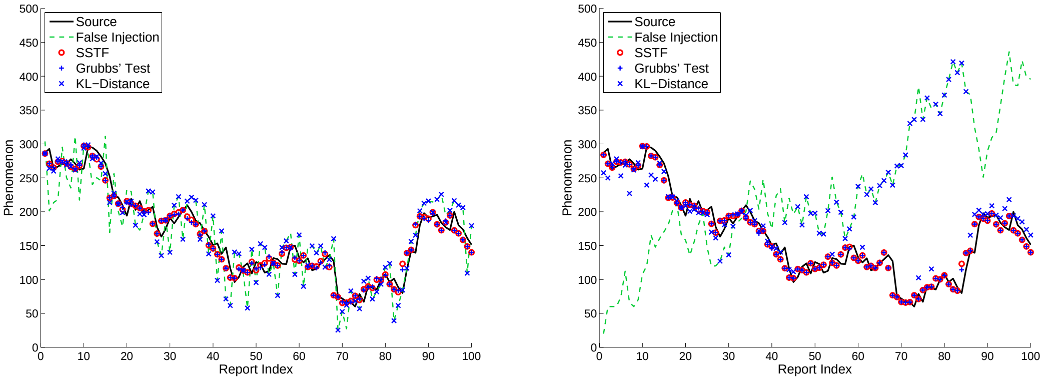
X-axis represents the standard deviation of the varying source data which is indicative of phenomenon variation. Y-axis represents the false injection. Z-axis represents the effect on r_{Ag} expressed as a percentage of the source data standard deviation. It can be seen that, for a constant false injection, the impact of false data increases with the phenomenon variation rate. On the other hand for a constant rate, as the false injection is increased, the impact of false data first increases and then decreases and becomes zero as the false injection is increased further. This is attributed to the fact that the false reading remains no longer a part of the largest bin and is excluded from aggregation. It is also observed that impact on r_{Ag} is very limited (up to 2% of source data standard deviation) which conforms to our security analysis (refer to Section 6.1).

7.2.3 Comparison with Other Schemes

To further demonstrate the effectiveness of SSTF in limiting the impact of false data, we compare our scheme with two recent works. The first scheme Yang et al. (2006) is an outlier detection scheme based on *Grubbs' test* Frank (1969) referred to as *Grubbs' Test outlier detection scheme*. In this scheme the cluster nodes report only the sensed readings. During the decision making process, CH calculates the mean of the received readings and computes a standard error of the readings about this mean to find the outliers. Then CH excludes the outliers and averages the remaining readings to generate the final aggregated reading.

The second scheme Zhang et al. (2006) is a reputation based scheme based on *KL-Distance technique* Cover and Thomas (1991) referred to as *KL-Distance reputation-based scheme*. In this scheme the cluster nodes report sensed readings only. However, CH forms a reputation for each of the sensors based on the history of readings received by computing KL-Distances. By applying a K-means clustering algorithm it forms groups and averages the readings from the group with highest reputation for generating the aggregated reading.

The results are shown in Fig. 7.4 and Fig. 7.5. Shown in each figure is a snapshot of a randomly varying source; the Y-axis represents the value in units of the phenomenon; the X-axis represents the report index and a stretch of 100 reports is shown. In Fig. 7.4 there is only one compromised node while in Fig. 7.5 there are four compromised nodes doing a colluded attack and we can see that SSTF



(a) Attack satisfying Bin Test

(b) Arbitrary Attack

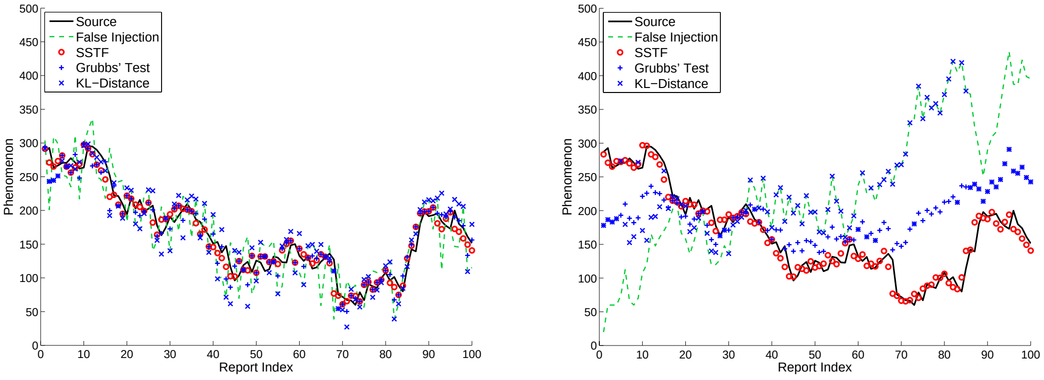
Figure 7.4 Demonstrating the effectiveness of SSTF in the presence of one compromised node. Shown are two attack strategies. In (a), the attacker chooses to inject false data that complies to the bin test and in (b), the attacker injects arbitrarily random false data.

is able to preserve the data transience and resembles the source data closely in both cases.

Fig. 7.4(a) shows the case when attacker does a subtle attack by reporting false readings that pass the distribution test and bin test. However, the impact on the aggregated reading is almost negligible.

In Fig. 7.4(b), the attacker injects arbitrarily random false data, which doesn't alter the result of our scheme either. This is because most of the injected false data has been excluded during the testing process. However it is seen in Fig. 7.4 that Grubbs' test outlier detection scheme can sometimes perform as good as our scheme. This happens because, when the number of compromised nodes is small, the false data could be identified successfully. KL-Distance reputation-based scheme performs poorly because the reputation over a history of readings enables even the false readings to pass the test.

In contrast, Fig. 7.5 shows the performance comparison of the schemes in the presence of four compromised nodes. In this case, our scheme still performs well due to the bin test, while the other two schemes fail significantly because four out of ten cluster nodes are compromised and perform a colluded attack. Since only the current readings are reported in these two schemes, the colluded false readings usually pass the test and get to participate in the aggregation, resulting in a prominently false reporting.



(a) Attack satisfying Bin Test

(b) Arbitrary Attack

Figure 7.5 Demonstrating the effectiveness of SSTF in the presence of four compromised nodes. Shown are two attack strategies. In (a), the attacker chooses to inject false data that complies to the bin test and in (b), the attacker injects arbitrarily random false data.

CHAPTER 8. Conclusions and Future Directions

One of the important security concerns in wireless sensor networks is false data injection attack wherein the adversary captures a sensor and starts injecting malicious data into the network. In the past, many schemes have been presented to deal with false data injection attacks. However, we observed and demonstrated that these schemes cease to give satisfactory performance in the presence of a source exhibiting time varying phenomenon. In most of the existing schemes sensors report a single value (only their current reading) to the cluster head for decision making purposes. This makes it difficult to judge whether the reading is genuine or false in the case of a time-variant phenomenon.

In this work, we provide statistical tools and techniques to overcome the limitations in existing schemes and give mathematical analysis and simulations to demonstrate the effectiveness of our proposed statistical framework. We show the modularity of the statistical framework and how it can be applied on a security scheme suitably enhanced to work with the framework. We evaluate the performance of our scheme in various phenomenon variation scenarios for varying percentages of adversaries in the network and observe that our scheme achieves its goal of minimizing the impact of false injection while preserving phenomenon variations.

To show a complete realization of our statistical framework, we present SSTF, a secure statistical scheme to distinguish data transience from false injection in clustered sensor networks. SSTF employs our developed statistical framework and we enhance the IHHAS scheme to be used as the underlying security framework. In contrast to existing false data filtering schemes, SSTF requires each individual sensor to report a statistical digest, in addition to the sensed reading and we emphasize the merits of this strategy to effectively monitor transient variations in the phenomenon. Through detailed theoretical analysis and extensive simulation study, we demonstrate the effectiveness of SSTF in preserving the transient data while being resilient to false data injection attacks.

Several important research directions emerge from my work. We can find more applications of our framework to design schemes to preserve false data injection in the presence of transient phenomenon in a dynamic topology sensor network or in a structure-free aggregation setup. We designed SSTF primarily for applications requiring periodic reporting and monitoring, it could be investigated into applying SSTF to query-based setup, wherein the sensors respond to a query from BS. We could also investigate the extension of our current idea to scenarios with a more general sense or aggregation such as network coding as part of the aggregation scheme. Further, our scheme could be applied to other diverse research topics such as mesh network security or spam filtering in the internet.

BIBLIOGRAPHY

http://www.intel.com/research/exploratory/instrument_world.htm Instrumenting the world. (Online link, last date accessed: November 26, 2007).

Buchegger, S. and Boudec, J.-Y. L. 2002. Performance analysis of the CONFIDANT protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). In *Proceedings of IEEE/ACM MobiHOC*.

Chen, B.-C., Chen, L., Ramakrishnan, R., and Musicant, D. R. 2006. Learning from aggregate views. In *Proc. ACM ICDE*.

Cover, T. M. and Thomas, J. A. 1991. *Elements of Information Theory*. John Wiley.

Eschenauer, L. and Gligor, V. D. 2002. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*.

Fan, K., Liu, S., and Sinha, P. 2006. *On the Potential of Structure-Free Data Aggregation in Sensor Networks*. Proc. INFOCOM'06, Barcelona, Spain.

Frank, G. 1969. Procedures for detecting outlying observations in samples. In *Technometrics*. Vol. 11. 1–21.

Ganeriwal, S. and Srivastava, M. B. 2004. Reputation-based framework for high integrity sensor networks. In *Proceedings of SASN '04*.

Hogg, R. 1983. *Probability and statistical inference*, 2 ed. Macmillan Publishing Co., Inc.

Hu, L. and Evans, D. 2003. Secure aggregation for wireless networks. In *Proc. Workshop on Security and Assurance in Adhoc Networks (WSAAN'03)*.

Janakiram, D., Reddy V, A. M., and Kumar, A. V. U. P. 2006. Outlier detection in wireless sensor networks using bayesian belief networks. In *Proc. IEEE Comsware*.

Liu, F., Cheng, X., and Chen, D. 2007. Insider attacker detection in wireless sensor networks. In *Proc. IEEE INFOCOM*.

Ma, J. and Perkins, S. 2003. Online novelty detection on temporal sequences. In *Proc. ACM SIGKDD*.

Mahimkar, A. and Rappaport, T. 2004. SecureDAV: A secure data aggregation and verification protocol for sensor networks. In *Proc. IEEE Global Telecommunications Conference (GLOBE-COM'04)*.

Michiardi, P. and Molva, R. 2002. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*.

Oliveira, A. L. I., Neto, F. B. L., and Meira, S. R. L. 2006. Improving novelty detection in short time series through rbf-dda parameter adjustment. In *Proc. IEEE International Joint Conference on Neural Networks*.

Przydatek, B., Song, D., and Perrig, A. 2003. SIA: Secure information aggregation in sensor networks. In *ACM SenSys 2003*.

Sheng, B., Li, Q., Mao, W., and Jin, W. 2007. Outlier detection in sensor networks. In *Proc. ACM MobiHoc*.

Shukla, V. and Qiao, D. 2007a. Distinguishing data transience from false injection in sensor networks. In *Proc. IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'07)*.

Shukla, V. and Qiao, D. 2007b. A robust statistical scheme to monitor transient phenomenon in sensor networks. In *Proc. IEEE International Conference on Communications (ICC'07)*.

Srinivasan, A., Teitelbaum, J., Liang, H., Wu, J., and Cardei, M. 2006. Reputation and trust-based systems for ad hoc and sensor networks. In *Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks, Wiley&Sons*.

Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. 2006. Online outlier detection in sensor data using nonparametric models. In *Proc. ACM VLDB*.

Wagner, D. 2004. Resilient aggregation in sensor networks. In *Proc. 2nd ACM workshop on Security of ad hoc and sensor networks (SASN'04)*.

Wang, Z.-Q., Wang, S.-K., Hong, T., and Wan, X.-H. A spatial outlier detection algorithm based multi-attributive correlation.

Wu, Q. and Shao, Z. 2005. Network anomaly detection using time series analysis. In *Proc. IEEE ICAS/ICNS*.

Yang, H. and Lu, S. 2004. Commutative cipher based en-route filtering in wireless sensor networks. In *Proc. IEEE Vehicular Technology Conference, (VTC'04)*.

Yang, Y., Wang, X., Zhu, S., and Cao, G. 2006. SDAP: A secure hop-by-hop data aggregation protocol for sensor networks. In *Proc. The ACM International Symposium on Mobile Ad Hoc Networking and Computing, (MOBIHOC'06)*.

Ye, F., Luo, H., Lu, S., and Zhang, L. 2004. Statistical en-route filtering of injected false data in sensor networks. In *Proc. IEEE INFOCOM*.

Yu, Z. and Guan, Y. 2006. A dynamic en-route scheme for filtering false data injection in wireless sensor networks. In *Proc. IEEE INFOCOM*.

Zhang, W., Das, S. K., and Liu, Y. 2006. A trust based framework for secure data aggregation in wireless sensor networks. In *Proc. IEEE SECON*.

Zhang, Y., Yang, J., and Vu, H. 2006. Interleaved authentication for filtering false reports in multipath routing based sensor networks. In *Proc. IEEE IPDPS*.

Zhu, S., Setia, S., Jajodia, S., and Ning, P. 2004. An interleaved hop-by-hop authentication scheme for filtering false data injection in sensor networks. In *Proc. IEEE Symposium on Security and Privacy, (SSP'04)*.