# Stony Brook University


OFFICIAL COPY

**The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.**

**© All Rights Reserved by Author.**

# A Resilient Actuation Attack on Wireless Sensor Networks

A Thesis Presented

by

**Aneeta Bhattacharyya**

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Master of Science**

in

**Computer Science**

Stony Brook University

**May 2010**

**Stony Brook University**

The Graduate School

**Aneeta Bhattacharyya**

We, the thesis committee for the above candidate for the

Master of Science degree, hereby recommend

acceptance of this thesis.

**Jennifer Wong – Thesis Advisor**

**Assistant Professor, Computer Science Department**

**Samir Das – Chairperson of Defense**

**Associate Professor, Computer Science Department**

**Jie Gao**

**Assistant Professor, Computer Science Department**

This thesis is accepted by the Graduate School

Lawrence Martin

Dean of the Graduate School

Abstract of the Thesis

**A Resilient Actuation Attack on Wireless Sensor Networks**

by

**Aneeta Bhattacharyya**

**Master of Science**

in

**Computer Science**

Stony Brook University

**2010**

Wireless sensor networks are built using tiny wireless sensor devices which have limited computational power and energy resources, as a result of which they can be subjected to various security compromises, including denial-of-service attacks. A particularly detrimental active denial-of-service attack that damages the sensing fidelity of wireless sensor networks is known as the *Actuation attack* [1], where hostile actuator (or actor) nodes belonging to a foreign network directly perturb or distort the environmental conditions being monitored. In this thesis we explore the loss of availability and reliability of wireless sensor networks ensuing from a proposed resilient Actuation attack which uses randomness and discontinuity to remain imperceptible. We demonstrate how the attack is designed to collect intelligence, without compromising any nodes or data packets, and to exercise caution to evade detection. We show how various factors, such as the frequency of actuation, topology of sensor network, forwarding scheme used by the network, and density of hostile nodes impact the efficacy of the attack. We discuss several possible techniques to defend against the proposed Actuation attack, and analyze their effectiveness. Finally, we conclude that it is increasingly difficult to control or detect an attack of this form owing to its random and asymmetric nature.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

**W**ireless **S**ensor **N**etworks (WSNs) are rising as a promising new technology to facilitate economically feasible solutions for a variety of applications such as military surveillance, forest fire monitoring, robot control, industrial automation, infrastructure protection, and habitat monitoring. These networks use a large array of resource-constrained sensor nodes with multiple preset functions, such as sensing and processing, to fulfill different application objectives. In the past, a de facto definition was established for WSN as a large-scale (thousands of nodes, covering large geographical areas), wireless, ad hoc, multi-hop, resource-constrained, un-partitioned network of homogeneous, tiny (hardly noticeable), mostly immobile (after deployment) sensor nodes that would be randomly deployed in the area of interest [5]. This characterization is mostly valid for applications that were used in the military domain. More recently other civilian application domains of WSNs have been considered, such as environmental and species monitoring, which have lead to a broader definition of WSN that includes heterogeneous and mobile sensor nodes, simple network topology, use of existing infrastructure (e.g., cellular phones), actuation capability, etc.

In the past few years wireless sensor networks equipped with actuation ability has quickly evolved as a topic of high interest [2]. These networks can not only sense their surroundings, but can also directly influence certain observable facts in their surroundings. As a result of their unique abilities, Wireless Sensor Actuator Networks (WSANs) may turn out to be an integral part of systems that facilitate microclimate control in buildings, battlefield surveillance, environmental monitoring, and nuclear, biological and chemical attack detection [6]. Unfortunately this novel paradigm also opens the doors to a new-fangled class of active and distributed attacks, known as *Actuation attacks*, that cripples the sensing fidelity through actuation [1], [2], [3]. An *Actuation attack* aims to alter data before it enters the WSN by physically distorting, or altering the phenomenon of interest. It results in a loss of network dependability and decreased lifetime without physically capturing any of the nodes, or compromising the security keys. Consequently, the mechanisms devised to protect against attacks on data inside the WSN as well as on routing and control data are ineffective. Furthermore, the attack is distributed in nature, potentially allowing any number of legitimate nodes to be victimized [1].

In this thesis, we model and analyze the impact of a resilient Actuation attack on the performance and lifetime of an immobile multi-hop WSN (topology of the network remains unchanged after deployment) that has been deployed to monitor a physical parameter such as temperature, humidity, pressure, level of oxygen in the air,

and so on. The objective of such an attack is to trigger huge amount of packet flow, thereby, using up essential network resources (power supply and bandwidth), and rendering it inoperable. We presuppose that the legal sensor nodes do no coordinate amongst each other, that the network uses controlled flooding to propagate packets and that only the base station performs data aggregation and analysis. Our proposed Actuation attack uses randomness and discontinuity to foil any attempts by the legitimate network to discover an attack. To reduce the chances of detection the adversary assaults only parts of the network at a time for short random intervals instead of attacking the entire network at the same time. The nodes being attacked at any point of time are also selected arbitrarily following no fixed order. We argue that due to the randomness incorporated into the attack the use of statistical tools, or internal data models will not help the legitimate network to determine which nodes are being attacked.

An attack of this form can become quite insidious, since the adversary has no need to capture communication packets, or break and use encryption/decryption keys. All she needs to do is to use actuation locally, and perturb the phenomenon being monitored without worrying about the data that is being transferred to the base station. So no amount of expensive and resource consuming encryption technology can protect against such an attack, as the data gets altered even before it is being measured. A key assumption that is made in the research of sensor networks is that only $k << N$ nodes can get captured by the adversary [1]. Because an Actuation attack is distributed in nature such an assumption is made invalid, as depending upon the distribution factor, the attacker could cover all, or most of the legitimate nodes in a network.

The attack is carried out by a malicious wireless sensor actuator network comprised of a number of malicious wireless sensor nodes (mWSN for short) which are distributed randomly throughout the same physical space as the legitimate sensor nodes (lWSN for short). The legal network is assumed to follow a data-driven event management system where nodes transmit packets only when the phenomenon being monitored deviates from the normal range of fluctuations. This is a reasonable assumption considering that the legitimate WSN is expected to minimize its energy expenditure to ensure longevity of operation. The adversary cashes in from this policy by using actuation to fluctuate the phenomenon beyond its *normal* range and triggering massive packet flow. Ideally, the mWSNs should be deployed in such a way that they are able to victimize nodes which are furthest away from the bases station. This would cause forwarding of packets by all intermediate nodes, thus resulting in power drainage from not only the victim nodes, but also the intermediate nodes. However, in the absence of any a priori knowledge of the locations of the legitimate nodes the adversary will have to choose random distribution because of which the attack will take longer to cripple the network, but it would be no less effective.

The aim of this thesis is to show how an Actuation attack can successfully use randomness and discontinuity to evasively attack and take down a wireless sensor network deployed to continuously monitor a spatially distributed physical phenomenon. The rest of the thesis is organized as follows. Chapter 2 provides some background, and discusses work in related area. In Chapter 3 we discuss various

facets of sensor network architecture that influence the design of the proposed attack. Thereafter, in Chapter 4 we describe several security issues related to WSNs, and the related topic of Actuation attack. Chapter 5 illustrates and analyzes the framework required for the proposed attack. We describe the details of the experimental setup required to model the attack in Chapter 6. We present the actual simulation results obtained in Chapter 7, and propose various possible countermeasures for the attack. Finally, we conclude in Chapter 8.

# Chapter 2

# Related Work

Actuation in WSNs in an emerging and largely unexplored topic with many open problems related to communication, coordination, reliability and security. Many researchers are currently investigating this new dimension of WSNs, and in this section we describe a set of related problems that have influenced our work. Section 2.1 presents a comprehensive study of research performed in the area of sensor network security that is pertinent to our research work. In Section 2.2, we address various research works related to actuation in WSNs. In Section 2.3, we overview different types of Actuation attacks, and some of the existing countermeasures for tackling Actuation attacks. Finally in Section 2.4, we address detection avoidance techniques that can be used by nodes to traverse through an unknown environment.

## 2.1 Security Challenges

As wireless sensor networks gain popularity, researchers are assigned the increasingly challenging task of making them secure and robust against all forms of attacks. Severe resource constraints, unreliable communication channels, uncontrollable and potentially harsh sensing environment, and unattended operations make it nearly impossible to implement traditional computer security techniques. In addition, the inability to secure the wireless medium proves to be an even bigger challenge. Therefore, researchers are forced to come up with new security approaches that specifically cater to the needs of wireless sensor network. Perrig et al. have developed a suite of security protocols known as SPINS for use in an extremely limited sensor network platform [14]. SPINS consists of two security building blocks – SNEP, a protocol for data confidentiality, two-party data authentication, and data freshness and μTESLA, a protocol that provides authentication for data broadcast. Their protocol suite relies on the concept that every node shares a secret key with the base station, which is at all times able to communicate with every node in the network. Schmidt et al. introduce a new security architecture for sensor networks that provides confidentiality, integrity, and authentication [10]. Even though their architecture cannot prevent capturing and compromising of nodes, it can minimize the effects of a captured node. In [17] the authors propose a novel location-based resilient security approach that overcomes the threshold limitation on the number of compromised nodes that a sensor network can handle, and provides graceful performance degradation against an increased number of compromised nodes.

In [8], [9], [15], [18], [19], and [20], the authors explore the challenges for security in wireless sensor networks, and classify many of the current security attacks along with enlisting the corresponding defensive measures. A critical factor in sensor network security is the issue of physical vulnerability of nodes deployed in an unattended and possibly hostile environment which poses extra security challenges that have not been fully addressed to date [2], [13]. The most common form of attacks on a WSN are Denial of Service (DoS) attacks; [7] and [11] specifically focus on these forms of attacks, and their corresponding prevention mechanisms. A widespread physical layer DoS attack is jamming attack which uses radio interference to exploit the shared nature of wireless medium, and prevents devices from communicating, or receiving [21].

## 2.2 Actuation in Wireless Sensor Networks

A WSN equipped with actuation ability can not only sense the physical world, but can also perform appropriate actions upon the environment. Sensor actuation includes, but is not limited to, actions such as turning on external fans (possibly to disperse heat, chemicals, or biological agents), and moving across the landscape (thereby, re-shaping the topology of the environment) [1]. In [6], Akyildiz and Kasimoglu present different types of WSANs (Wireless Sensor & Actor Networks), and investigate the coordination and communication problems at various network layers that arise in such networks due to the coexistence of sensors and actors. They suggest a new protocol stack to be specifically used for WSNs and WSANs that would consist of three planes – communication plane, coordination plane, and management plane.

Mobility has been proposed as a useful extension to sensor networks, as they add flexibility and dependability. In [22] the authors explore how mobility can be used in a sensor network to repair the coverage loss in the area being monitored. When a section of a network becomes non-functional the authors propose using self-aware actuation to allow the network to reorganize its available resources, and form a new functional topology in the face of run-time dynamics (called “self-aware” approach). Reference [23] deals with the issue of trustworthiness in large-scale low-energy wireless sensor networks. The authors have proposed a low-complexity transmission reliability scheme that is based on local wireless path repair, and hop-to-hop retransmissions. To protect from active attacks they have developed two-level re-keying/re-routing schemes that not only adapt to a dynamic network topology, but also securely update keys for each data transmission.

## 2.3 Actuation attacks and Countermeasures

In [1], [2], Czarlinska and Kundur present a general class of Actuation attacks which aim to disable the sensing fidelity and dependability of a wireless sensor

network. They propose random mobility as a possible countermeasure to such a Denial of Service on Sensing (DoSS) attack, and show how mobility can reduce the number of affected nodes exponentially under various deployments, network densities, and actuation radii. As an extension of their work, in [3] they model and assess the vulnerability of a WSN to an Actuation attack carried out by a hostile WSAN belonging to a foreign network. The attack is modeled to affect the decision that a WSN node reports about the presence, or absence of a phenomenon to its cluster head, and the work focuses on determining the probability that the WSN cluster head becomes alerted to such an attack given some statistical information about the phenomenon. In reference [24], Czarlinska and Kundur investigate the strength and stealth properties of Actuation attacks on event-driven visual sensor networks (VSNs). They probe the achievable actuation of hostile nodes that are not globally coordinated, and establish that given certain conditions, local optimization will result in a stronger stealthy attack than the global coordination case.

Many researchers have been working on identifying and/or preventing bogus sensing reports that can be injected by one, or more compromised nodes. SEF [27] is a statistical en-route filtering technique that can be used to detect and drop false sensing reports during the forwarding process. Authenticating event reports requires that nodes share certain security information; however, attackers can obtain such information by compromising just a single node. To overcome this limitation, SEF design divides a global key pool into multiple partitions, and carefully assigns a certain number of keys from one partition to individual nodes. Given that any single node knows only a limited amount of system secret, compromising one, or a small number of nodes cannot disable the overall network from detecting bogus reports. SEF design harnesses the advantage of large-scale by requiring endorsement of an event report from multiple detecting nodes, and by detecting false reports through collaborative filtering of all forwarding nodes along the path.

## 2.4 Detection Avoidance

When an adversary plans on perpetrating an Actuation attack on a victim sensor network her first objective becomes the deployment of the hostile sensor network avoiding any kind of detection. In [25] the authors formulate an efficient and effective algorithm for finding the minimum exposure path in sensor networks. By exposure they imply a measure of how well an object moving on an arbitrary path can be observed by the sensor network over a period of time. The algorithm works for arbitrary sensing and intensity models, and provides an unbounded level of accuracy as a function of runtime. Simulation results show that the algorithm can produce high quality results efficiently, and can be used as a performance and worst-case coverage analysis tool in sensor networks.

Remaining elusive while navigating to a goal in a dynamic environment containing an observer requires taking advantage of opportunistic cover as it occurs. Reference [26] presents a reactive navigation approach that allows stealthy traverses in unknown environments containing dynamic objects. The key is to take benefit of

coverage opportunities as they occur, particularly since a dynamic object offering beneficial coverage may only persist briefly. The proposed algorithm allows a robot/mobile node to reactively hide behind a mobile object, and dynamically adjust its position according to the movement of the object.

In [11] Howard, Matarić, and Sukhatme propose and evaluate an incremental greedy-algorithm to deploy nodes of a mobile sensor network into an unknown environment one-at-a-time, with each node making use of information gathered by previously deployed nodes to determine its target location. The algorithm is designed to maximize network coverage whilst simultaneously ensuring that nodes retain line-of-sight with one another. Results of simulation experiments show that the algorithm is able to achieve 70% to 85% coverage when nodes are made to select free space location that maximizes the *coverage heuristic* (location at which nodes cover the greatest area of presently unknown space). Furthermore, the algorithm scales as a polynomial function of the number of deployed nodes ($n$), and in the worst case of order of $n^2$.

These studies of stealthy deployment and detection avoidance help us to understand how an intruding malicious WSN can deploy itself in an environment undetected.

# Chapter 3

# Wireless Sensor Network Architecture

Wireless sensor networks have found their way into a variety of applications and systems with vastly varying requirements and characteristics due to their flexibility, cost-effectiveness, accuracy, and ease of deployment. In this section we discuss those dimensions of a wireless sensor network that directly, or indirectly affect the design of the proposed Actuation attack. Developers frequently design sensor networks to collect and analyze low-level data from an environment of interest. Accomplishing the network's goals depends on cooperation between individual sensor nodes, data aggregation, and data processing. The sensor nodes are small inexpensive wireless devices that are capable of sensing, local processing, and communicating wirelessly. However, they are constrained in terms of bandwidth, memory, energy, and computational power; so each node is capable of performing only a limited amount of processing. But when coordinated with the information from a bigger set of other nodes, they have the capacity to measure a given physical environment in great details [28]. So generally a large number of sensors are deployed into the sensing environment in the hope of achieving high sensing fidelity and acceptable coverage. Due to the nature of their operation sensor nodes are mostly required to work in unattended remote geographic location. This implies that the sensing environment can be dynamic, potentially harsh, uncontrollable, and untrustworthy. So maintaining data availability and freshness becomes a tough job.

There are several desirable functionalities of a sensor which include, but are not limited to: ease of installation, self-indication, self-diagnosis, reliability, time awareness for coordination with other nodes, some software functions and DSP, and standard control protocols and network interfaces [19]. Typically a sensor node is built using five basic components: sensing hardware, processor, memory, power supply (usually battery), and transceiver. They may also have additional application dependent components such as location finding system, power generator, and mobilizing module [30]. Owing to the need for deploying a large number of sensor nodes, the cost of a single node is very important to justify the cost of the entire network. As pointed out in [30], the cost of a single sensor node should be much less than $1 (USD) in order for the sensor network to be feasible. For this reason, most sensor nodes have no built-in tamper resistant mechanism that could guard against physical manipulation by an adversary.

WSNs typically include one or more base stations (or sink nodes, or gateways) that have enormously more computational power, energy, and communication resources than an ordinary sensor. They are adept to communicating with several sensors via radio links. Data is first recorded at the sensor nodes and then processed, compressed, and transmitted/forwarded to the base station(s). The transmitted data is

presented to the system (end user) by the gateway connection which collects, analyzes, and presents the measured data. We can view the base station as an interface between the users and the network. To retrieve information of interest from the network users need to submit queries and gather results from these base stations.

After the initial deployment (usually ad hoc), the sensor nodes are capable of self-organizing themselves into an appropriate network infrastructure in order to accomplish their appointed tasks, without any external guidance, or supervision [10]. Usually, the nodes are organized into a flat logical layout with no hierarchy amongst themselves, so that all devices are equal in terms of the role they can play in the network. Information is accumulated based on the sensing capabilities of these nodes, and the network makes decisions based on this gathered information.

The mode in which sensor nodes and bases station(s) communicate with one another is defined by the communication network. *Infrastructure-based networks* and *ad hoc networks* are two widespread forms of communication networks [5]. In infrastructure-based communication networks, sensor nodes can only directly communicate with the base stations. It is the job of the base station(s) to relay communication between individual sensor nodes. If there are multiple base stations then they must be able to communicate amongst each other. Mobile phone network is a type of infrastructure-based communication network. This type of communication model is more likely to be used in a sensor network when an infrastructure is already available. In ad hoc communication networks, nodes can directly communicate with one another without the help of base station(s). The nodes may act as routers, forwarding messages over multiple hops on behalf of other nodes. An ad hoc network is preferred in most applications, as it does not necessitate the use of expensive infrastructure.

Energy consumption is the most important factor to determine the life of a sensor network, as most sensor nodes are driven by battery, and have very low energy resources [28]. The radio subsystem (transceiver) generally requires the largest amount of power. Therefore, it is advantageous to transmit data over the radio network only when required. Such an *event-driven collection model* requires an algorithm to be loaded into all the sensor nodes to determine when to send data based on the sensed event [29]. As opposed to the event-driven working mode, in a *continuous working mode* the nodes transmit data non-stop over the radio network for as long as their batteries last. When the network has to operate for long periods of time the event-driven working mode is more suitable; whereas, when the network has to operate for a short period of time the continuous working mode is a more appropriate choice.

# Chapter 4

# Security Issues and Actuation

As sensor networks continue to grow, so does the need for effective security mechanisms. Security in sensor networks has a number of challenges, some of which are: wireless communication among the nodes, lack of pre-existing infrastructure, dynamic topology changes, and resource constraints in terms of memory, energy, and communication bandwidth [15]. Because sensor networks may interact with sensitive data and/or operate in potentially harsh, uncertain, and dynamic environments, it is imperative that these security concerns be addressed from the beginning of the system design. However, due to inherent memory, processing and computing constraints, security in sensor networks poses different challenges than traditional network security.

In Section 4.1 we detail the various security objectives of a sensor network. Thereafter, in Section 4.2 we provide details of various possible attacks on sensor networks. We focus on sensor actor nodes, and their mode of operation in Section 4.3. Finally, in Section 4.4 we discuss the details of Actuation attack.

## 4.1 Security Requirements

A sensor network is a special type of network. Although it shares some commonalities with a typical network, it also poses unique requirements of its own which arise due to resource constrictions, unattended operations, and unreliable communication. Thus, the security requirements of WSNs cover both the typical network requirements and the unique requirements suited solely to wireless networks.

The security objectives in sensor networks have been summarized below [8, 15]:

- *Data Confidentiality*: In many applications WSNs are required to gather highly sensitive and classified information. Data confidentiality ensures that this data is protected, and will not be leaked to unauthorized parties. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that is possessed by only the intended receiver.
- *Data Authentication*: This requirement allows the receiver of a data/communication packet to verify that the packet was really sent by the node it claims to be coming from. Data authentication can be achieved by using a Message Authentication Code (MAC) on the communicated data.

- *Data Integrity*: This requirement ensures that the data has not been altered, or modified by an adversary while in transit. Data loss or damage can also occur due to harsh communication environment.
- *Data Freshness & Availability*: In many cases sensor networks are required to monitor time-sensitive events. So it is important to ensure that the data provided by the network is fresh and available at all times. This proscribes an adversary from carrying out a replay attack in the future. To ensure data freshness a time-related counter is added into the packet.
- *Survivability*: This requirement ensures that the network is able to provide a minimum level of service in the presence of power loss, failure and attacks; and its performance degrades gracefully when a small portion of the nodes are compromised.
- *Self-Organization*: In case of ad-hoc WSNs the nodes should have the independence and flexibility to self-organize and self-heal themselves according to changes in situations. Without self-organization capability, the damage resulting from an attack or even from hazardous environment may be devastating.

## 4.2 Attacks in Wireless Sensor Networks

Sensor networks are particularly vulnerable to several key types of attacks mainly due to the broadcast nature of wireless communication and the lack of tamper resistant hardware. Furthermore, limited resources and computational power imply that public key cryptography cannot be used as a viable security solution. Adversaries can perpetrate attacks in a variety of ways, most notably as denial-of-service attacks, besides traffic analysis attacks, node replication, attacks against privacy, Sybil attack, and physical attack.

Denial of service attacks on wireless sensor networks can range from simply jamming the sensor's communication channel to more sophisticated attacks designed to violate the 802.11 MAC protocol or any other layer of the wireless sensor network [12]. Due to potential asymmetry in power and computational constraints, it is nearly impossible to guard against a well orchestrated DoS attack. Commonly used techniques for preventing against DoS attacks include payment for network resources, pushback, strong authentication and identification of traffic. In case of node replication attacks, an attacker seeks to add a node to an existing sensor network by copying the ID of an existing sensor node [8]. A node replicated in this fashion can severely disrupt a sensor network's performance: packets can be corrupted, or misrouted, cryptographic keys can be compromised, and in the worst case part of the network can be disconnected altogether.

Another particularly harmful attack on sensor and ad hoc networks is known as the *Sybil attack* where a malicious node illegitimately claims multiple identities using the identities of other legitimate nodes [4]. Besides defeating distributed data storage systems, the Sybil attack is also effective against routing algorithms, data aggregation, voting, fair resource allocation, and foiling misbehavior detection.

Another challenge is the harsh outdoor environment in which most sensor networks are required to operate which gives way to various kinds of physical attacks. Unlike many other attacks mentioned above, physical attacks can destroy sensor nodes permanently, so the losses are irreversible [8]. For instance, attackers can extract cryptographic keys, tamper with the associated circuitry, modify programming in the sensors, or replace them with malicious sensors under the control of the attacker.

Reflecting upon the different security challenges and possible attacks on sensor networks, four key issues have been identified for providing security to wireless sensor networks which have been outlined below [9]:

- *Key management*: Providing key-management service in WSNs poses significant difficulty due the ad-hoc nature of the environment, intermittent connectivity, resource limitation, limited connectivity, etc.
- *Encryption & Decryption mechanism*: Due to constraint on memory and processing power it is not possible to use asymmetric cryptographic techniques for encryption and decryption.
- *Secure Routing*: In the absence of secure routing attackers may successfully inject erroneous routing information, alter routing information, or compromising a node to broadcast malicious information to the base station.
- *Prevention of Denial-of-Service*: Any event that diminishes, or eliminates a network's capacity to perform its expected function will be considered to be a DOS attack [7].

## 4.3 Wireless Sensor Actor Nodes

Recent technological advances have lead to the emergence of distributed Wireless Sensor Actuator (or Actor) Networks (WSANs) which are capable of observing the physical world, processing the data, making decisions based on observations, and performing appropriate actions [6]. A WSAN usually consists of a set of sensor nodes, having low-energy and limited mobility, and an additional set of higher-energy actuator nodes [3]. The sensor nodes collect information about the physical world, while the actors take decisions based on this collected information and then perform appropriate actions upon the environment, which allows a user to effectively sense and act from a distance. As shown in Figure 4.1, the sensor and actor nodes are scattered in the sensor/actor field while the sink monitors the overall network and communicates with the task manager node as well as the sensor/actor nodes. Sensors detecting a phenomenon either transmit their readings to the actor nodes which process all incoming data, and initiate appropriate actions, or route data back to the sink which may issue action commands to the actors. The former is known as *Automated Architecture* while the latter is known as *Semi-Automated Architecture* [6].

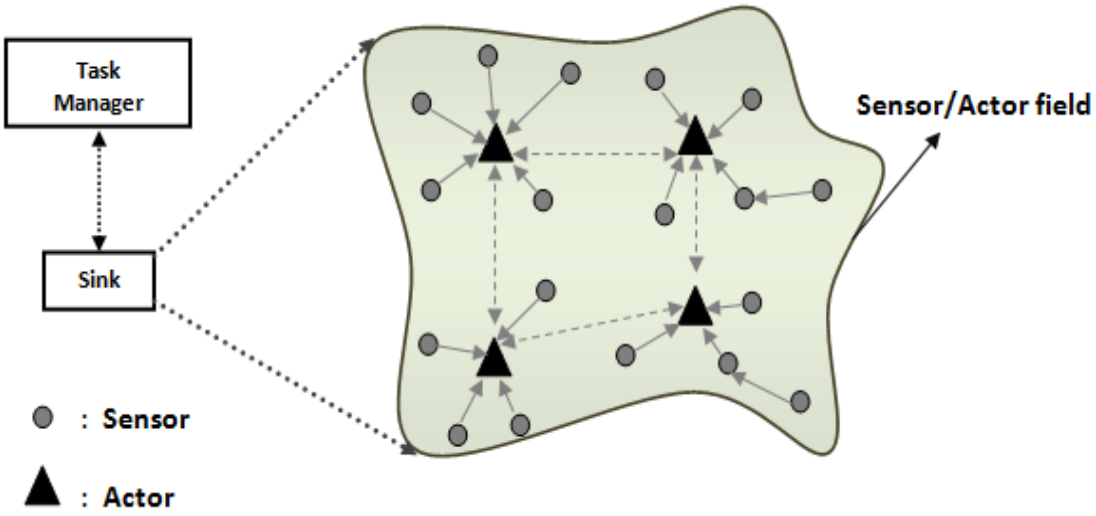



Figure 4.1: The **Physical Architecture of WSAN**: As detailed in [6], the nodes are scattered in the sensor/actor field while the sink monitors the overall network.

## 4.4 Actuation attack

The emerging paradigm of actuation in WSNs is bringing with itself several questions related to security and reliability. A potential security issue that we are particularly interested in is the likelihood of the actuation process maliciously affecting sensor readings registered by the network nodes. Since actuation affects sensing directly at the physical level of data collection, protection mechanisms relying on data encryption occur too late for the attack to be averted [3], as can be seen from Figure 4.2 and Figure 4.3. An attack of this form has been referred to as *Denial-of-Service on Sensing* (*DoSS*) attack because the sensing fidelity of the legitimate network is compromised by the malicious actuator nodes [1, 2]. This form of distributed attack is different from other active attacks in that the malicious nodes are not attacking the data inside the communication packets. Rather they are actuating the phenomenon being monitored by the legitimate nodes, and hence forcing them to report false intelligence about the environment to the central base station.

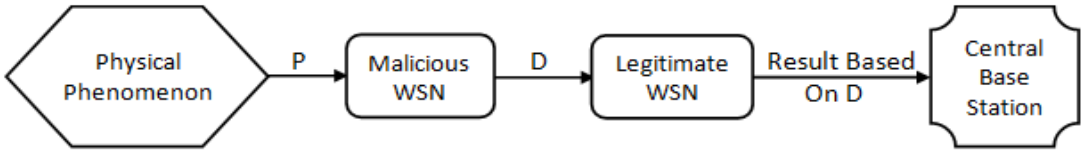



Figure 4.2: Flow of information during an Actuation attack

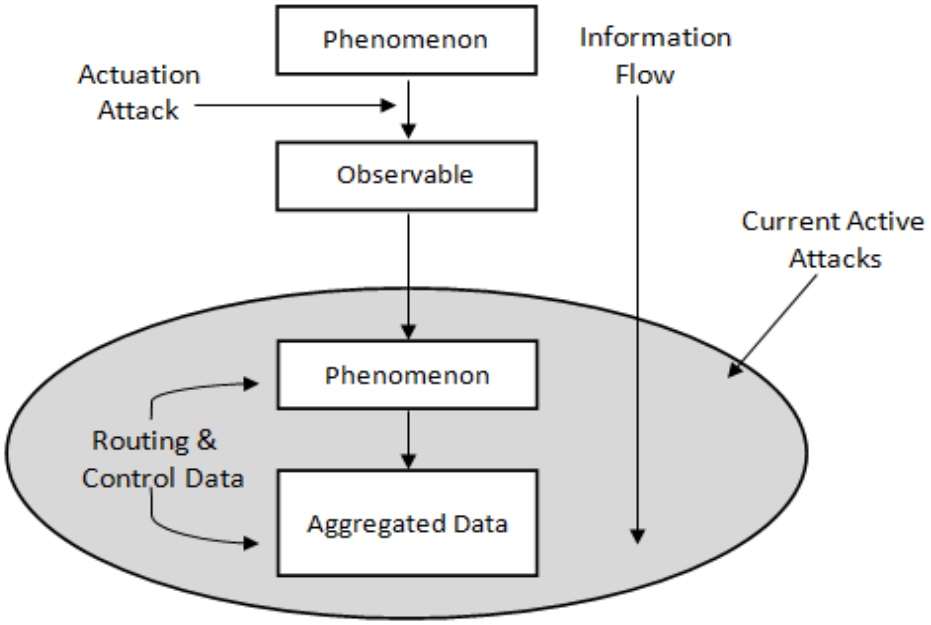



Figure 4.3: Actuation vs. other active sensor network attacks as detailed in [1]

# Chapter 5

# Framework

The proposed Actuation attack targets WSNs that have been set up to continuously monitor distributed physical phenomenon such as temperature, wind, humidity, light, sound, level of pollution, etc. With such networks it is possible to alter the event being measured leading to increased network traffic, and ultimately failure of victim nodes. Measurements can be altered by means of lighting a fire, using a fan, creating shade, generating sound, etc. This puts the attacker on the advantage if her main goal is to consume network bandwidth and to take down the network. In Section 5.1 we provide a description of the legitimate system and in Section 5.2 we give a detailed description of the malicious system.

## 5.1 Legitimate System Description

We have considered a sensor network in which each node is capable of measuring its surrounding phenomenon of interest, and comparing it against a predefined *expectation model*. If it finds an observed value deviating from the theoretical value, it writes its observations into a data packet, and transmits it to the central administrator. The central administrator may be located a few levels higher than the sender node. Accordingly, the data packet will have to be forwarded via the intermediate nodes to be received by the central administrator. This is how communication happens with the base station whilst using multi-hop routing algorithm.

### 5.1.1 Network Setup

The legitimate network is formed using $N$ homogenous static sensor nodes distributed randomly throughout a finite region to observe a spatially distributed physical phenomenon of interest. A single base station is placed at the centre of the network for data aggregation and network maintenance. We presume that the base station has been supplied with infinite battery power. All nodes have identical transmission radii (unit disk graph), and have the same amount of initial battery energy. Also, the cost of sending/receiving packets is the same for all nodes.

### 5.1.2 Event-Driven Data Management

We assume that there is a short bootstrapping phase right after the initial network setup, in which the base station supplies each node with a local expectation model of the phenomenon being observed. A node uses this expectation model to test each of its observations. When an observed value does not tally with the expectation model, the node creates a data packet containing the observed value, and transmits/forwards it to the base station. No operation takes place when the observed value matches the expectation model. In this way, the network is able to conserve energy, and operate for a longer period of time.

### 5.1.3 Expectation Model

In order to build an expectation model, the base station first collects data recordings from all sensors for the first few days. This is achieved by directing the nodes to transmit data packets only once at the end of the day containing all the observations made that day. Once the base station receives all recordings made by a node (say $p$), it plots a simple time vs. data curve, and performs regression model analysis to calculate the local average variation in readings for that node over the course of time. This local average variation is used as an estimate of error threshold by node $p$, and is denoted by $\varepsilon_p$. After completing estimate calculations for all nodes the base station hands out the error threshold values to the corresponding sensor nodes. If at any time node $p$ observes a difference in readings between two consecutive observations to be greater than $\varepsilon_p$, it generates a data packet containing its current data recordings, and transmits it to the base station. No packets are sent out for readings that fall within the error threshold.

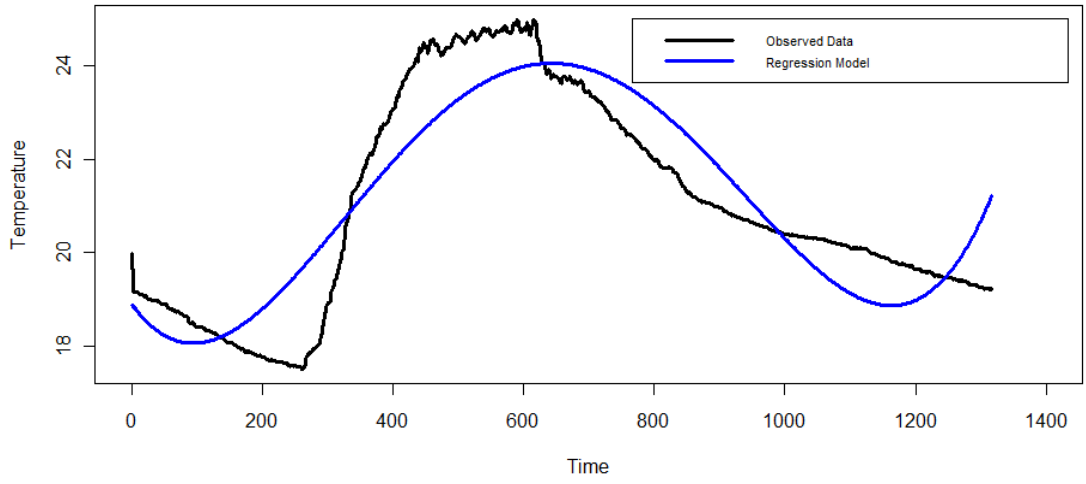



Figure 5.1: **Least Square Model**: Temperature vs. Time curve for a single node along with linear least square regression model used to calculate error threshold

### 5.1.4 Multi-hop Communication Model

We assume a simple communication model in which each sensor node has a transmission radius of $R$ over which it can communicate to reach its neighboring nodes. Additionally we assume that the communication radius of each node is much smaller than the total area of the network. For multi-hop communication to be possible it is necessary that $R$ be sufficiently large, so that the connectivity of the nodes is maintained. The nodes follow a directional flooding scheme in which packets are selectively forwarded to only those neighbors which have minimum hop count to the base station. As flooding decisions are based on the direction towards the central administrator, nodes acquire information about themselves, their neighbors, and the base station, using a location detection system such as the GPS.

The base station, on the other hand, has complete knowledge of the entire network topology. It may either use a shortest-path algorithm to transmit packets to specific sensor nodes, or use broadcasting techniques to transmit packets to all nodes. The base station performs data aggregation and analysis as well as replies to queries from end users based on the observations made by the network.

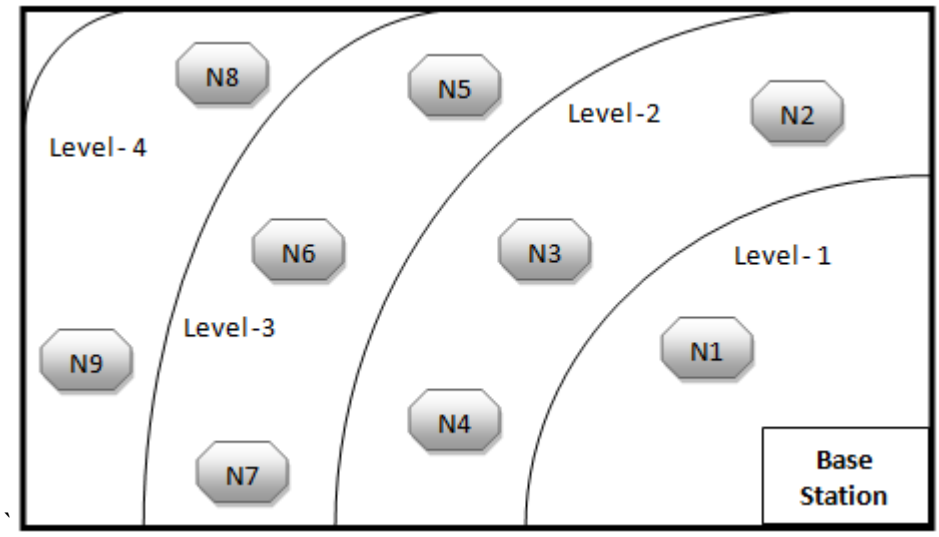



Figure 5.2: Subset of nodes along with the base station of a legitimate network

We visualize the entire network to be divided into concentric circles, with the base station being the centre. All nodes that are 1-hop away from the base station form the level-1 circle; all nodes that are 2-hops away from the base station form the level-2 circle; and so on.

### 5.1.5 Network Assumptions

We mentioned some of the assumptions that have been made for the legitimate network in the previous sections. Below we summarize all the assumptions made for the legitimate network so far:

- The network consists of $N$ homogeneous immobile sensor nodes with identical transmission radius $R$ and the ability to continuously monitor a spatially distributed phenomenon of interest.

- There is a single base station in the network which physically resides in the centre of the network, and possesses complete knowledge of the entire network topology.

- All nodes have globally unique IDs, and every node knows the IDs of all its neighbors.

- Post deployment the network topology remains static.

- The time and energy costs for sending and receiving a packet of unit size are the same.

- All transmissions from any node $u$ are Omni-directional. Thus, any message sent by $u$ can be received by any node in its *neighborhood* (within the node's transmitting range). The neighborhood of node $u$ is denoted by $N(u)$.

- The network uses certain power management scheme by which packets are transmitted only when there is a major variation in reading from the expected model.

- The nodes do not use any coordination amongst themselves to analyze data.

- The base station has the ability to cut off a node from the network when it suspects the node to be under attack. This decision can be based on a heuristic limit on acceptable change in phenomenon. If a node starts to behave erratically and reports changes in phenomenon beyond the acceptable limit, the base station shall force the node out of the network.

## 5.2 Malicious System Description

In the beginning of this section we provide an overview of the proposed attack. Thereafter, we describe in details the attack model and its required settings. Subsequently, we show how Gaussian distribution can be used to build a smarter attack. A discussion of the assumptions made for the attack setup concludes this section.

### 5.2.1 Attack Overview

Actuation in sensor networks is defined as the ability of a node to act upon, change, or influence its environment using limited energy [1], [2], [3]. When in the wrong hands, actuation could be employed to coerce nodes into misrepresenting the phenomenon being monitored by a WSN. We are primarily focusing on WSNs that use some kind of power management scheme that permits packets to be transmitted only when there is a sudden unexpected variation in readings. For such networks, it is possible to use malicious actuator nodes to disturb the phenomenon under observation, and force legitimate nodes to send out unnecessary data packets. We call the data packets that are spawned due to actuation *false event reports*. Effectiveness of the Actuation attack depends on the number of false event reports that can be forcefully generated. False event reports have a twofold effect. First, they contaminate the process of data analysis at the base station by supplying tainted data. Second, they use up valuable battery life of all nodes that are compelled to transmit/forward these packets. We are interested in investigating the latter effect of these packets.

The fact that a single change in observation causes multitude of data transmissions enables an attacker to carry out her attack by targeting only a limited number of nodes in the network. When activated the malicious actor nodes trigger a sudden radical change in the surrounding phenomenon, as a result of which readings deviate from the expectation model. If executed effectively, the large amount of communication would consume essential network bandwidth and drain battery power, ultimately resulting in the shutdown of the total network. This happens as a result of the fact that sensor nodes have very limited power which they can harvest, or store.

A few things need to be taken into account while designing a smart and efficient attack. Firstly, it is not a good idea to cause deviation in readings of a specific node for a prolonged period of time. This is because the base station has the mechanism to cut off a specific node from the network if it starts transmitting a huge number of *unexpected* data packets. Such a mechanism is enforced into a sensor network to conserve energy, and to check potential attacks. Secondly, the attacker may not have the means to target all nodes of a huge commercial network (sensor nodes are deployed in very large numbers), so she will have to use her resources effectively to cause maximum damage.

The first challenge can be handled by attacking a node for a short span of time, and then shifting to a different node. The victim nodes will report sudden unexpected changes in observations only for a short interval of time. By keeping the time span of

attack short, the adversary will be able to evade the consequence of shutting down of victim nodes. The second challenge can be tackled by deploying the malicious nodes as widely as possible, so as to cover the maximum area. If we imagine the network to be divided into sub-regions then the attacker should try to deploy at least one malicious node per sub-region. The size of the sub-region would depend upon the number of malicious nodes at the disposal of the attacker.

### 5.2.2 Actuation attack Model

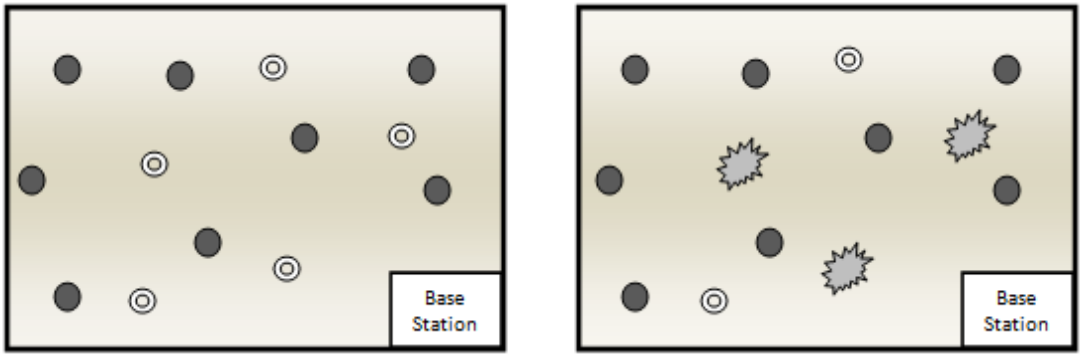
While modeling the attack, we assume that the attacker has a wireless sensor network at her disposal that possesses actuation capability using which the environmental condition being monitored by the legitimate network can be altered. The malicious network is formed using *M* homogeneous static actor nodes, where M will depend on the resources available to the attacker. These actors are also capable of sensor reporting like ordinary sensor nodes. For our attack model we have assumed M << N in order to analyze the effectiveness of the attack with limited resources. The attacker will deploy these *M* nodes evenly into the sensing environment, so that they are likely to affect the maximum possible number of lWSNs. In order to avoid detection, the mWSNs operate independently and perform no communication with each other. Also, there is no requirement for a base station.

For an attack to proceed, the malicious network needs to be deployed into the sensing environment without first getting detected by the legal network. To attain this, the malicious network could be deployed into the environment before the legitimate network is deployed, or it may be deployed alongside the legitimate network before the latter establishes its infrastructure and begins monitoring [1], [2]. Additionally, the adversary may choose to use a detection avoidance algorithm as discussed in Section 2.4, in order to deploy the malicious network into the sensing environment without being discovered.

In our model, all mWSNs are fitted with self-timers which can be set to random counts. Once deployed each node sets its self-timer to a random value, and waits for the timer to expire. During this waiting period it records the surrounding phenomenon pretty much the same way that a legitimate node does. The recorded information is used to devise a more intelligent actuation process as we describe in the next section. When the timer expires, the malicious node initiates an Actuation attack which affects all legitimate nodes in its neighborhood. The span of attack is kept short and random. At the end of the attack the malicious node resets its self-timer to another random value, and the entire process is repeated continuously.

As a result of random timer values and random attack spans, the set of malicious nodes assaulting the legitimate network at any point of time is unpredictable which helps to avoid detection due to symmetry. The set of victim lWSNs in the neighborhood of the activated mWSNs send out false event reports to the base station. Because attack period of each mWSN is short and independent of other mWSNs, the likelihood of alerting the base station is minimized. The unpredictable attack procedure is continued until the legitimate sensor network gets flooded with false event reports that drain away node energy, and eventually force the network to shut down under the burden. We note that, as a result of Omni-directional transmissions

by the lWSNs, the mWSNs are able to receive all data packets that are being transmitted in their neighborhood, which help them in estimating the amount of damage that is being caused.




(a) (b)

: lWSN
: (inactive) mWSN
: (active) mWSN

Figure 5.3: **Random Discontinuous Actuation:** (a) Subset of legal nodes and 5 inactive malicious nodes that have been deployed into the same sensing environment. (b) Three of the malicious nodes attacking the network after being activated at some point of time.

### 5.2.3 Actuation Using Gaussian Distribution

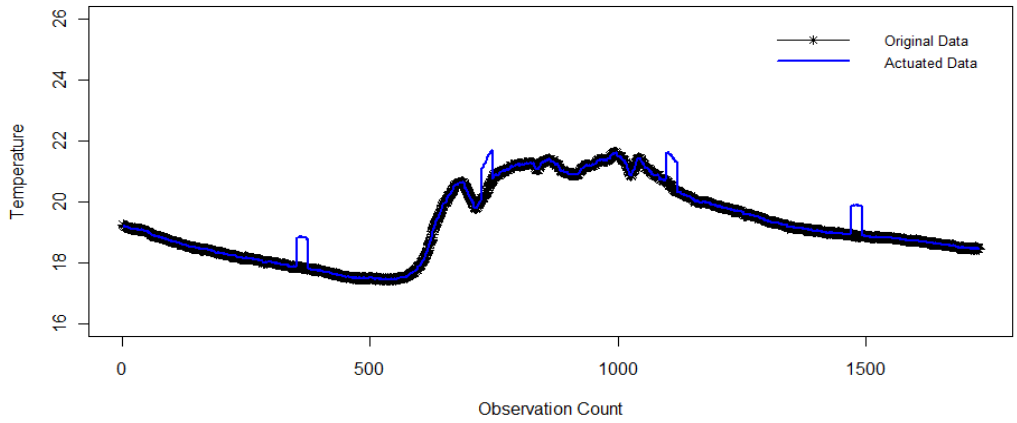
As specified in the previous section, when not attacking, the hostile mWSNs record their surrounding phenomenon just like the legitimate nodes. When the timer expires, the recorded observations are used to fabricate an actuation based on Gaussian distribution. A Gaussian function is a function of the form:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad (1)$$

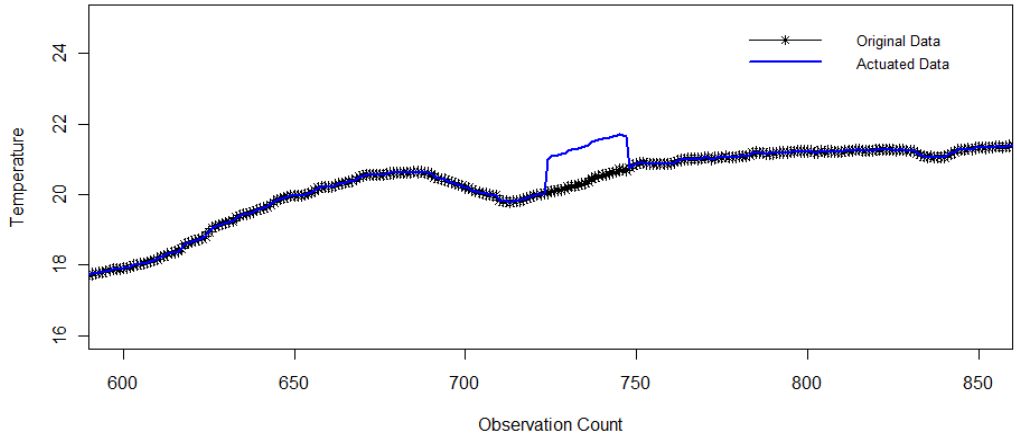
for some real constants $a > 0$, $b$, $c > 0$, and $e \approx 2.718281828$ (Euler's number). The graph of a Gaussian function is a characteristic symmetric "bell curve" shape that quickly falls off towards plus/minus infinity. The parameter $a$ is the height of the curve's peak, $b$ is the position of the centre of the peak, and $c$ controls the width of the "bell" [32].

The recoded observations provide an approximation on the amount of fluctuation that can be caused without alerting the base station. When a malicious node decides to initiate an attack, it first calculates a random span for which the attack should be sustained. It then checks its recordings made at the same time the previous day, and

calculates the average variation between consecutive observations for the span of time the attack will be sustained. We use $\alpha$ to denote the average variation in consecutive observations. A predefined constant value $\theta$ is added to $\alpha$ to set the height of the Gaussian curve. The constant value $\theta$ can be derived using some heuristics, or can be based on the number of false event reports that are being generated. The resulting bell curve is used by the malicious node to control the amount of actuation that it will cause to the phenomenon being observed.




(a)




(b)

Figure 5.4: **Actuation Peaks:** (a) Temperature vs. Observation Count Plot of original data recorded by an lWSN, and the actuated data generated by an mWSN. (b) A section of plot (a) zoomed in to show how the actuated peak deviates from the original data.

### 5.2.4 Deployment Assumptions

We have made a number of assumptions regarding the deployment and operation of the malicious wireless sensor network which we summarize below:

- The malicious sensor network is constructed using $M$ homogenous static actor nodes, where $M << N$.
- Using appropriate means, the malicious network is deployed without being detected by the legitimate network.
- A malicious node cannot masquerade itself as a legal member of the legitimate network, and send its own information, or replay old data.
- The mWSNs are capable of measuring the physical phenomenon of interest, just like the lWSNs.
- The mWSNs can sense all data packets that are being transmitted in their neighborhood. But they cannot compromise the data packets to read the information inside.
- Once deployed the topology of the malicious network remains static.
- Adversary has no knowledge of the topology of the legitimate network, so she cannot target specific nodes (for e.g., leaf nodes) to cause more damage.
- No communication takes places amongst the mWSNs, as it may lead to detection.

# Chapter 6

# Experimental Setup

For our experiments we have focused on sensor networks that are used to measure temperature. This section details the prototype design and setup used for our evaluation. We describe the original network from which the simulation model has been derived in Section 6.1. Section 6.2 presents the implementation of the simulation model.

## 6.1 Sensor Network Used

In order to simulate a legitimate network that records temperature, we have made use of data collected from a sensor network deployed in the Intel Berkley Research lab by the database group at MIT from 28th February to 5thApril, 2004 [31]. 54 sensor nodes were used to construct this network, as show in Figure 6.1. *Mica2Dot* [33] sensors with weather boards collected time-stamped topology information, along with humidity, temperature, light and voltage values once every 31 seconds. Data was collected using the *TinyDB* in-network query processing system [34], built on the *TinyOS* platform [35]. We are simulating our legal network based on the temperature recordings made by these nodes for a period of 19 days, from 28th February to 17th March, 2004.

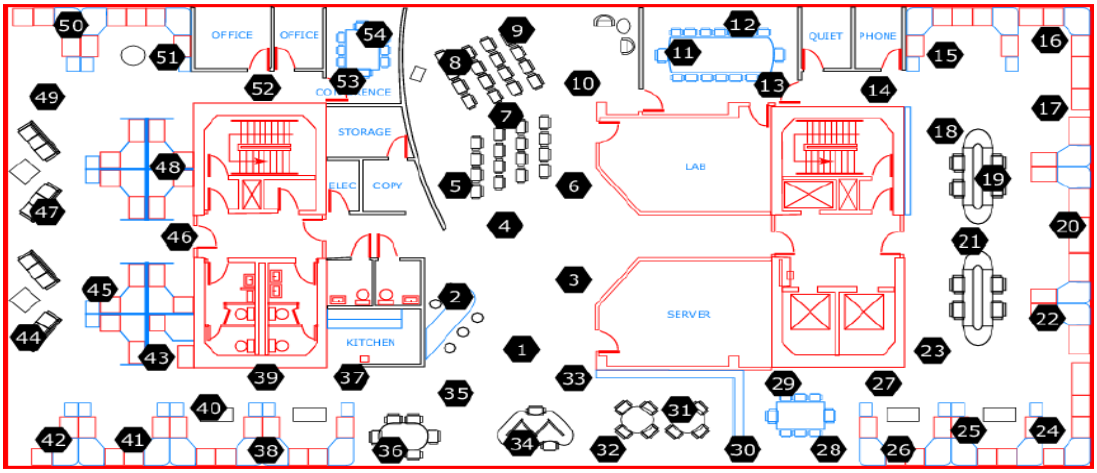



Figure 6.1: Arrangement of sensors in the Intel Berkley Research Lab as detailed in [31].

We have used a subset of these 54 nodes, namely 1 through 35, to constitute our legitimate network. For the purpose of simulation we have divided these 35 nodes into 3 hierarchical levels as shown in Figure 6.2. The original network does not detail any communication between the individual nodes, nor does it specify the existence of a base station. But our legitimate system model requires at least one base station as well as a packet forwarding scheme between nodes. Hence, we have assumed that a base station is present at the centre of the network, and all nodes transmit/forward data packets to this base station.

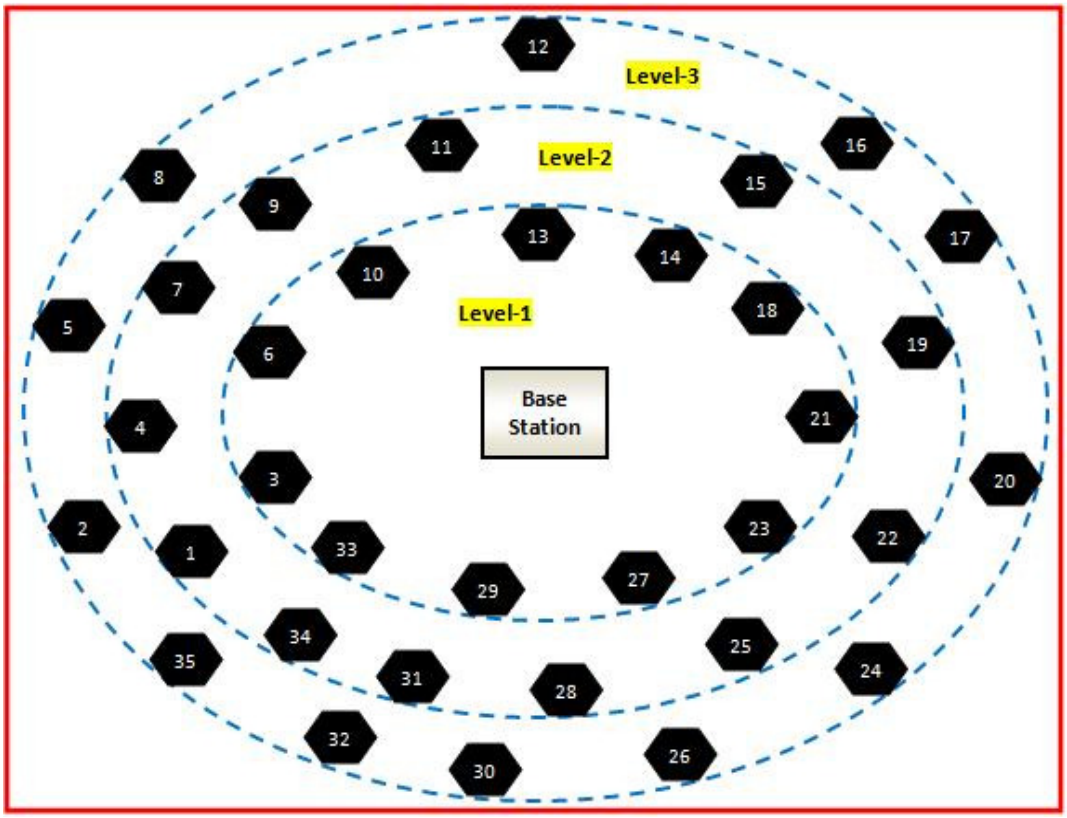



Figure 6.2: **Topology of Legitimate Network for Simulation**: *Level-1* comprises of nodes 3, 6, 10, 13, 14, 18, 21, 23, 27, 29, and 33. *Level-2* comprises of nodes 1, 4, 7, 9, 11, 15, 19, 22, 25, 28, 31, and 34. Finally, *Level-3* is formed using nodes 2, 5, 8, 12, 16, 17, 20, 24, 26, 30, 32, and 35.

The nodes in *Level-1* are assumed to be 1-hop away from the base station, so they can directly transmit their packets to the base station; nodes in *Level-2* are assumed to be 2-hops away from the base station, and are required to forward their packets to neighbors in *Level-1*; finally nodes in *Level-3* are assumed to be 3-hops away from the base station, and need to forward their packets to neighbors in *Level-2*. Thus, we establish a directional flooding scheme for all nodes. As mentioned in Section 5.1.4,

the sensor nodes follow a directional flooding scheme in which packets are selectively forwarded to only those neighbors which have minimum hop count to the base station. In our simulation, for each lWSN, we have indentified the neighbors with the minimum hop count to the base station, and we call those neighbors the *forwarding neighbors* of that node. Table 6.1 lists the *forwarding neighbors* of all 35 legitimate sensor nodes. Each node has the knowledge of its forwarding neighbors; whenever it has to transmit a packet, it checks its list of forwarding neighbors, and transmits the packet to only those nodes. This selective forwarding can be easily accomplished by including a list of receiver IDs with each packet. When a node has to transmit a packet it includes a list of its forwarding neighbor IDs with the packet, and broadcasts it. When a node receives a packet it checks the list of receiver IDs to see if it is on the list; it forwards the packet if its ID is found on the list, otherwise, no action is taken, and the packet is dropped.

| **Sensor Node** | **Forwarding Neighbor** | **Sensor Node** | **Forwarding Neighbor** |
|---|---|---|---|
| 1 | 3, 33 | 19 | 18, 21 |
| 2 | 1, 4 | 20 | 19, 22 |
| 3 | *Base Station* | 21 | *Base Station* |
| 4 | 3, 6 | 22 | 21, 23 |
| 5 | 4, 7 | 23 | *Base Station* |
| 6 | *Base Station* | 24 | 22, 25 |
| 7 | 6, 10 | 25 | 23, 27 |
| 8 | 7, 9 | 26 | 25, 28 |
| 9 | 10 | 27 | *Base Station* |
| 10 | *Base Station* | 28 | 27, 29 |
| 11 | 10, 13 | 29 | *Base Station* |
| 12 | 11 | 30 | 28, 31 |
| 13 | *Base Station* | 31 | 29, 33 |
| 14 | *Base Station* | 32 | 31 |
| 15 | 14, 18 | 33 | *Base Station* |
| 16 | 15 | 34 | 33 |
| 17 | 19 | 35 | 1, 34 |
| 18 | *Base Station* | | |

Table 6.1: **Forwarding Neighbors:** Legitimate sensor nodes, and their corresponding *forwarding neighbors.*

## 6.2 Simulation Model Development

The simulation environment was developed using Microsoft Visual Studio 2008, and R, a language and environment for statistical computing and graphics [36]. The data files from [31] were filtered to obtain the date, time, and temperature readings for each of the 35 nodes. We have assumed uniform power consumption for packet transmission that is independent of node location. The battery life of a node $n$ is symbolized by an integer counter $C_n$. During the initialization phase, $C_n$ is set to 500 for all $n \in N$. Each time a node transmits a packet, the counter is decremented by 1 to represent a single unit of power consumption. Eventually when the value of $C_n$ becomes 0 the corresponding node $n$ is declared to be dead.

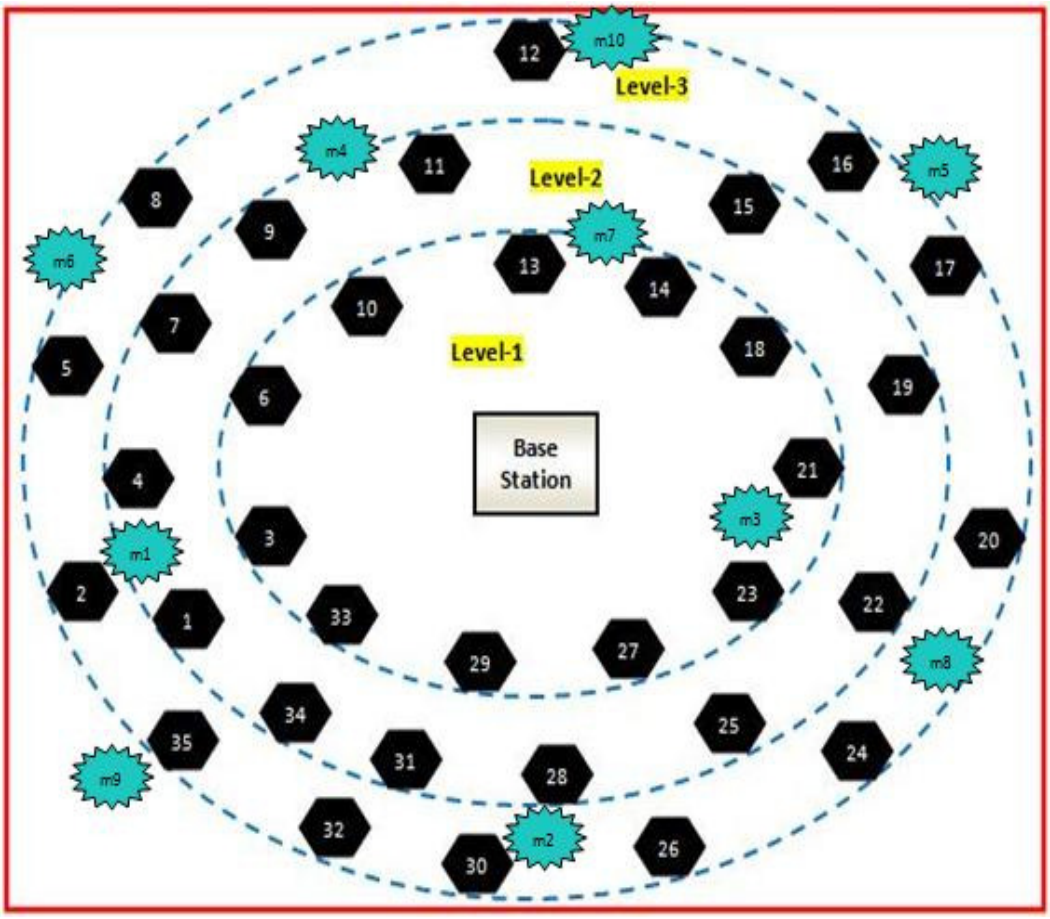



Figure 6.3: **Malicious Node Layout**: Malicious node *m1* affects the observations of nodes 1, 2, and 3; *m2* affects readings of 28 and 30; *m3* affects nodes 21 and 23; *m4* affects nodes 9 and 11; *m5* affects nodes 16 and 17; *m6* affects nodes 5 and 8; *m7* actuates the readings of nodes 13 and 14; *m8* affects nodes 20, 22, and 24; *m9* affects the readings of node 35; finally, *m10* actuates the readings of node 12.

In the attack simulation we have used 10 actor nodes to put together our malicious network. These nodes are distributed randomly throughout the sensing environment, which has been detailed in Figure 6.3. A random number generator is used by each malicious node to set its self-timer, and to determine the span of each attack. When the timer expires, an attack is initiated for the decided amount of time. As mentioned in Section 5.2.3, the attacks are based on Gaussian distribution function. Without a loss of generality we have set the values of parameters $a = \alpha + \theta$, $b = 0$, $c = 5$, $e =$ 2.718281828, in Eq. 1. The value of parameter $x$ is varied between -2.0 to 2.0. The value of parameter $\alpha$ is calculated from previous day's observations. Value for parameter $\theta$ is deduced through trial and error: we begin with a very small value of $\theta$, and keep introducing bigger values till a sudden increase in packet transmission is sensed. Using this method, the temperature is actuated in a controlled fashion which significantly reduces the chances of alarming the base station. This step is decisive in ensuring that the attack remains undetectable to the legitimate network. Once an actuation concludes, the corresponding node generates a new random number, and sets its timer to repeat the entire process again. This process is continued recurrently until the legitimate network is rendered inoperable. The battery consumption of malicious nodes has not been accounted for, since energy constraint is unlikely to prevent an Actuation attack from happening [2]. While a legitimate WSN is expected to minimize its energy expenditure to ensure longevity of operation, the goal of the malicious network may be a direct short-lived attack after which it can stop operating. Furthermore, the malicious actuator nodes are not required to operate (actuate the environment) continuously, hence they can survive longer than a legitimate node, which has to operate at all times.

# Chapter 7

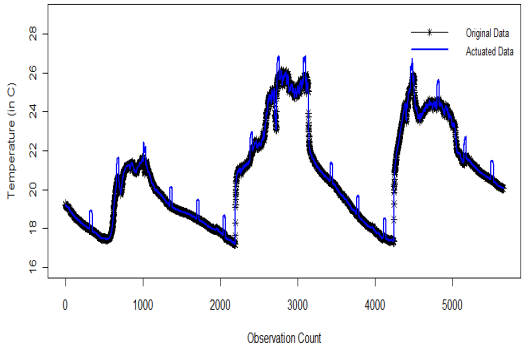
# Simulation Results & Insight

In this section we present simulation results to study the effect of our proposed Actuation attack which uses randomness and discontinuity to remain elusive and untraceable. Specifically, we examine the consequence on victim wireless sensor nodes, and the corresponding increase in false event reports. Temperature recordings of 19 days, from February 28$^{th}$ to March 17$^{th}$, were used for the experiments. The number of lWSNs was held constant at 35, while the number of mWSNs was varied from 1 to 10 to study the effectiveness of increase in malicious node density. Figure 7.1 shows the actuation of temperature caused by malicious nodes *m1*, *m6*, and *m9* on the readings of legitimate nodes 1, 2, 3, 4, 6, and 33 for a period of 3 days, from February 29$^{th}$ to March 2$^{nd}$. Since the malicious network is not synchronized, the actuation caused by each mWSN becomes independent of one another. Moreover, a single actuation is sustained for a very short span of time, and multiple such actuations are perpetrated over the given period of time. We have used two different groups of random values to set the self-timers of the mWSNs. In the first group random numbers ranging between (300, 600) have been used, and the set is called *random timer set I*. In the second group random numbers ranging between (200, 400) have been used, and the set is called *random timer set II*. Section 7.1 describes the efficacy of increasing malicious node density, while Section 7.2 explains the effectiveness of increasing the frequency of attack. Subsequently, in Section 7.3 we discuss some possible countermeasures for the given attack, and their usefulness.

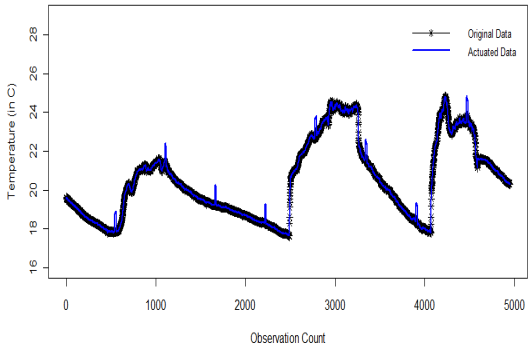
## 7.1 Effect of increasing malicious node density

We begin by looking at the efficacy of increasing malicious node density on packet transmission rate of individual lWSNs. For this purpose, we worked with a small subset of lWSNs that consisted of nodes 1, 2, 3, 4, 6, and 33. Any subset of the network could have been used for this purpose. We investigated the effect of malicious nodes *m1*, *m6*, and *m9* on these lWSNs by activating them one after the other. As can be seen from Figures 7.2 and 7.3, with the inclusion of each mWSN the rate of packet transmission of lWSNs in the neighborhood of the activated mWSNs increases manifolds. Alongside, the rate of packet transmission of forwarding neighbors of victim nodes also increases. For example, malicious node *m1* not only victimizes its neighbors 1, 2, and 4, but also affects the packet transmission rates of nodes 3, 6, and 33, since they happen to be the forwarding neighbors of the victim nodes. Whenever a node is forced to transmit/forward more than 500 packets, it uses

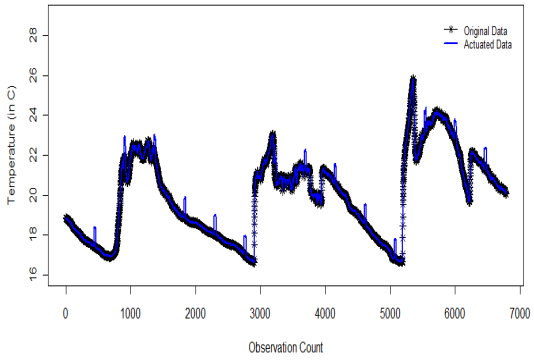
up all its power supply and stops functioning. All three mWSNs operating together cause nodes 3, 6, and 33 to die in the first case (Figure 7.2), and nodes 1, 3, 4, 6, and 33 to die in the second case (Figure 7.3). This shows how severe the attack can be when the adversary has all the required resources (in this case a sufficient number of malicious actor nodes).



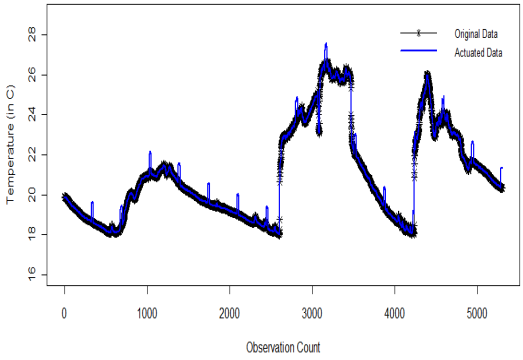

(a) Node 1



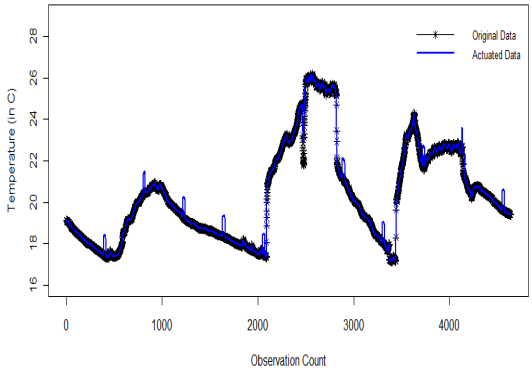

(b) Node 2



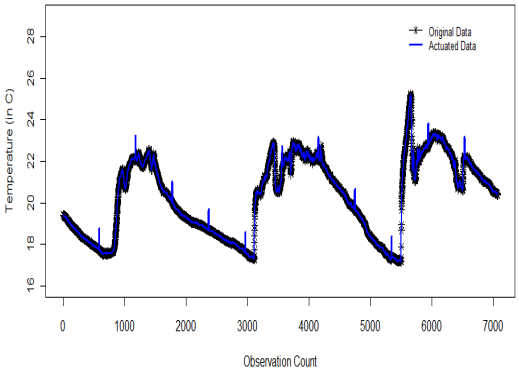

(a) Node 3




(b) Node 4

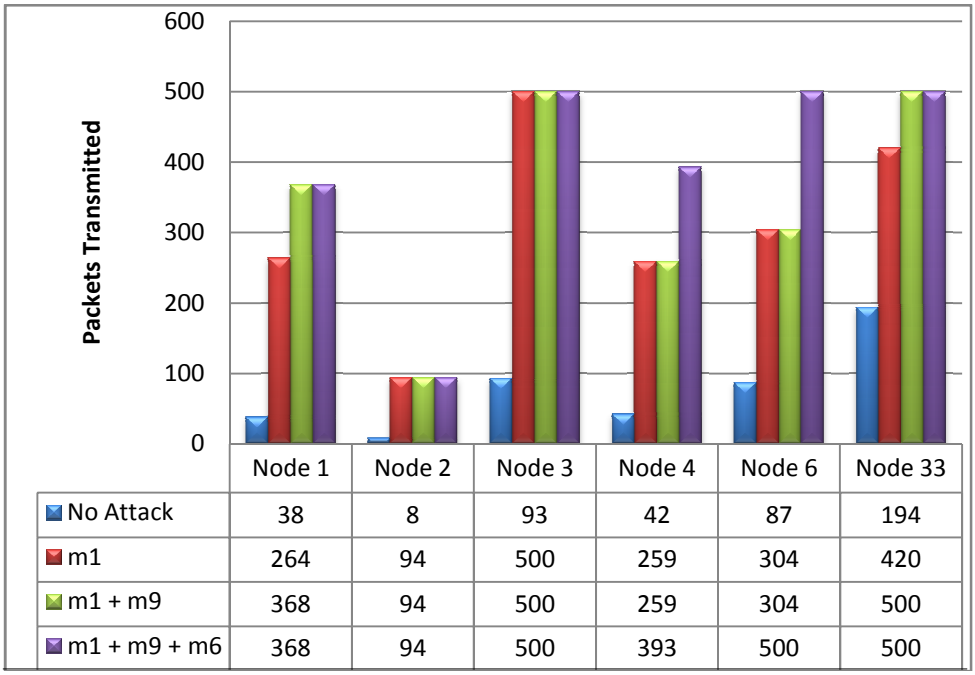



(a) Node 6



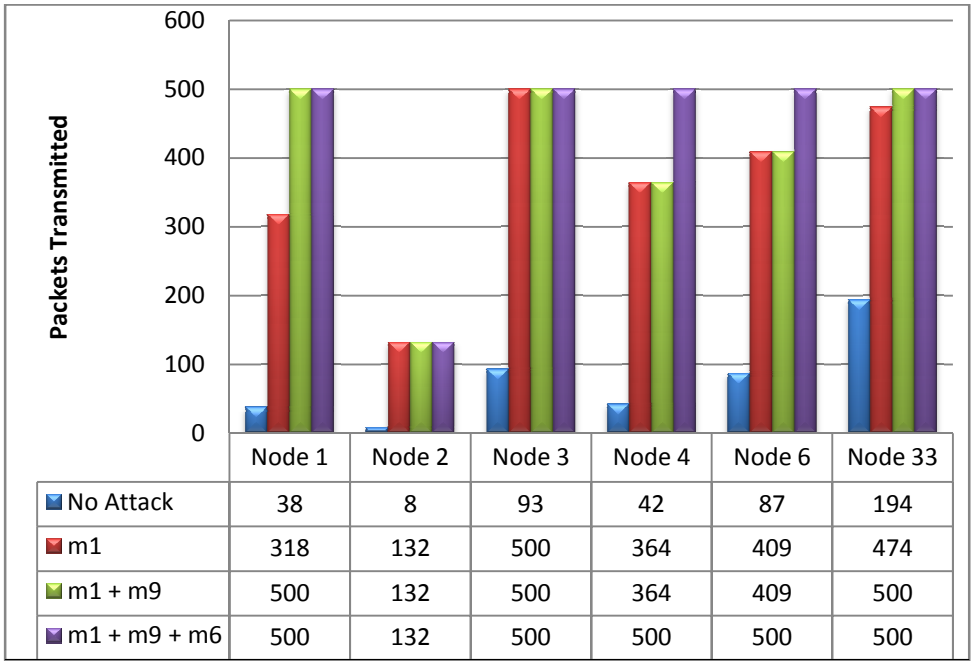

(b) Node 33

Figure 7.1: **Actuation of Readings**: Temperature has been measured in °C, and observations have been taken every 31 seconds for 3 days. Timers of mWSNs have been set using *random timer set I*.




| | Node 1 | Node 2 | Node 3 | Node 4 | Node 6 | Node 33 |
|---|---|---|---|---|---|---|
| No Attack | 38 | 8 | 93 | 42 | 87 | 194 |
| m1 | 264 | 94 | 500 | 259 | 304 | 420 |
| m1 + m9 | 368 | 94 | 500 | 259 | 304 | 500 |
| m1 + m9 + m6 | 368 | 94 | 500 | 393 | 500 | 500 |

Figure 7.2: **Individual Node Attack for Case 1**: Outcome of attack on a subset of legitimate nodes when *random timer set I* has been used to set self-timers of mWSNs.




| | Node 1 | Node 2 | Node 3 | Node 4 | Node 6 | Node 33 |
|---|---|---|---|---|---|---|
| No Attack | 38 | 8 | 93 | 42 | 87 | 194 |
| m1 | 318 | 132 | 500 | 364 | 409 | 474 |
| m1 + m9 | 500 | 132 | 500 | 364 | 409 | 500 |
| m1 + m9 + m6 | 500 | 132 | 500 | 500 | 500 | 500 |

Figure 7.3: **Individual Node Attack for Case 2:** Outcome of attack on a subset of legitimate nodes when *random timer set II* has been used to set self-timers of mWSNs.

Next we investigate the overall increase in the rate of packet transmission through the entire network. The malicious network layout detailed in Figure 6.3 is followed, but instead of activating all malicious nodes at the same time, we deploy one node at a time, starting with node *m1*, and moving through nodes *m2*, *m3*, *m4*, *m5*, *m6*, *m7*, *m8*, *m9*, and finally deploying node *m10*. This way we are able to judge the effect of increasing malicious node density on the rate of packet transmission as well as on the number of lWSNs rendered terminal. Table 7.1 and Table 7.2 show the total number of packets generated in the network, and the list of lWSNs that were rendered dead due to the inclusion of each malicious node for *random timer set I* and *random timer set II*, respectively.

| Malicious Node Introduced | Packets Generated | Dead lWSNs |
|---|---|---|
| None | 2,756 | None |
| m1 | 4,135 | 3 |
| m2 | 5,176 | 3, 29, 33 |
| m3 | 5,431 | 3, 29, 33 |
| m4 | 5,859 | 3, 29, 33 |
| m5 | 6,534 | 3, 29, 33, 21 |
| m6 | 7,557 | 3, 29, 33, 21, 6, 10 |
| m7 | 7,609 | 3, 29, 33, 21, 6, 10 |
| m8 | 8,608 | 3, 29, 33, 21, 6, 10, 23, 27 |
| m9 | 9,151 | 3, 29, 33, 21, 6, 10, 23, 27 |
| m10 | 9,211 | 3, 29, 33, 21, 6, 10, 23, 27 |

Table 7.1: **Attack outcome for Case 1**: Result of simulated attack on the network when *random timer set I* has been used to set self-timers of mWSNs.

| Malicious Node Introduced | Packets Generated | Dead lWSNs |
|---|---|---|
| None | 2,756 | None |
| m1 | 4,491 | 3 |
| m2 | 5,518 | 3, 29, 33 |
| m3 | 5,915 | 3, 29, 33, 21 |
| m4 | 6,703 | 3, 29, 33, 21 |
| m5 | 7,801 | 3, 29, 33, 21, 18 |
| m6 | 8,786 | 3, 29, 33, 21, 18, 4, 6, 10 |
| m7 | 8,889 | 3, 29, 33, 21, 18, 4, 6, 10 |
| m8 | 10,038 | 3, 29, 33, 21, 18, 4, 6, 10, 23, 25, 27 |
| m9 | 10,780 | 3, 29, 33, 21, 18, 4, 6, 10, 23, 25, 27, 1 |
| m10 | 10,870 | 3, 29, 33, 21, 18, 4, 6, 10, 23, 25, 27, 1 |

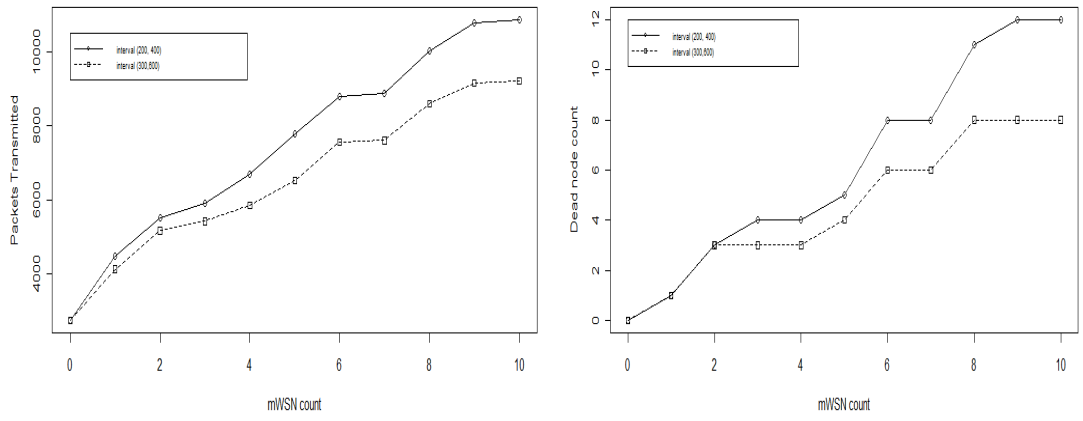
Table 7.2: **Attack outcome for Case 2**: Result of simulated attack on the network when *random timer set II* has been used to set self-timers of mWSNs.

When there are no mWSNs present in the system the total number of packets generated by the network over the course of 19 days is 2,756. With the inclusion of a single malicious node (*m1*) the packet count is increased by 1.5 times in case of *random timer set I*, and by more than 1.6 times in case of *random timer set II*. We see that a single mWSN results in the generation of 1379 and 1735 false event reports, respectively. When five of the mWSNs (*m1*, *m2*, *m3*, *m4*, *m5*) are deployed into the network, the packet count increases by 2.4 times in the first case, and by 2.8 times in the second case. Moving on, when all 10 mWSNs are deployed into the network, the packet count increases by 3.4 times in the first case, and by 3.9 times in the second case. Note that power consumption is directly proportional to the number of packets generated. Hence, as the count of false event reports increases, so does the number of dead nodes in the system. With a single malicious node in the system only one node was rendered dead (*node 3*) in both cases. When there were 5 mWSNs in the system 4 out of 35 nodes were rendered dead in the first case, and 5 out of 35 nodes were rendered dead in the second case. Eventually, when all mWSNs were deployed into the system, the damage caused in the first case was 8 dead nodes, and in the second case it was 12 dead nodes. These results clearly demonstrate that as the size of the malicious network increases, its coverage increases, and consequently the effectiveness of the Actuation attack increases.

## 7.2 Effect of increasing attack frequency

Next we examine how increasing attack frequency affects the severity of the proposed Actuation attack. Frequency of attack is controlled by the self-timers used by all mWSNs. By setting a smaller value to the timer we can effectively increase the number of actuations caused by each mWSN. Since lWSNs collect temperature information every 31 seconds, frequency has been based on the number of observations made by a node instead of time, as the number of observations directly relates to the passage of time. As mentioned before, we have simulated two cases, one using *random timer set I*, and the other using *random timer set II*, to study the effectiveness of attack frequency. *Random timer set II* causes actuations at a higher frequency than *random timer set I*. From Figure 7.4(a) we can infer that as the frequency of attack is increased, the number of packets generated in the network is also increased substantially. Similar deductions can be made by studying Tables 7.1 and 7.2, and comparing the '*Packet Generated*' column from both tables. We see that malicious node *m1* triggers 356 more false event reports when *random timer set II* is used. Likewise, when all malicious nodes are employed, they succeed in generating 1659 more event reports in the second case than in the first case.

Additionally, from Figure 7.4 (b), we can infer that an increase in number of actuations results in an increase in the number of dead nodes in the network. This can also be deduced from Tables 7.1 and 7.2. Thus, we have established that by increasing the frequency of actuation we can effectively increase the severity of the attack. But we note that the frequency of actuation cannot be increased to an infinite amount, since that may give away the presence of the malicious network. Our basic motive is to remain undetectable which can only be achieved by causing limited asymmetric disruption.




(a) (b)

Figure 7.4: **Outcome of increasing attack frequency:** (a) Plot for packet transmission vs. mWSN count with two different random timer sets. (b) Plot for dead node count vs. mWSN count with two different random timer sets.

## 7.3 Possible Countermeasures

We have worked on several possible countermeasures for preventing the proposed attack, but none have completely succeeded in averting the attack. Nonetheless, they can prove to be helpful in designing a successful countermeasure in the future, and for this reason we discuss our approaches in this section.

In our first approach we tried to use a statistical alert system where each node used an error window to judge whether it is under attack. This error window was based on the node's error threshold value $\varepsilon$, which was used for building its expectation model. Whenever an observed temperature fell beyond the defined error window, the corresponding node concluded that it was being attacked. As a modification to this method, we introduced the use of slope values, instead of exact temperature values, for attack detection. A node would compute slopes of readings taken the previous day in sets of 3, 4, or 5 to provide an estimate of temperature variation. Unfortunately such statistical alert tests cannot serve as true attack detectors, as they are prone to generating too many false positives, and can only be used to provide an indication to the possibility of an attack. This is because it is always possible that the network is witnessing a rare natural occurrence as a result of which temperature readings are not abiding by the expectation model.

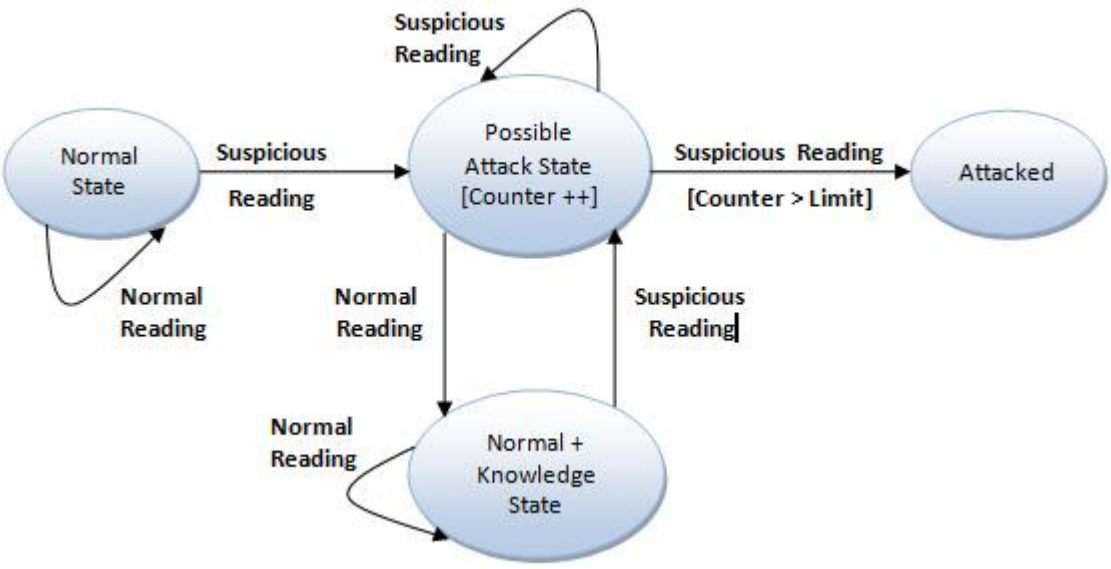



Figure 7.5: **Node State Transition Diagram:** A node can be in one of four possible states: *Normal State*, *Possible Attack State*, *Norma + Knowledge State*, and *Attacked State*. The states are changed based on current temperature recordings. When a node reaches the '*Attacked*' state it immediately alerts the base station about the same.

To avoid false positives we devised a state transition system for each lWSN, as shown in Figure 7.5. The state of all nodes is initially set to *Normal State* and a *limit* is set on the number of attack suspicions that the node would tolerate before alerting the base station. The limit value is critical in determining the effectiveness of this defensive measure, and is determined heuristically. Each node uses the aforementioned statistical alert system to check if its current phenomenon recording is "normal" or "suspicious". When the node encounters a suspicious observation it immediately moves itself into the *Possible Attack State*. In this state, it increments a counter which keeps tab of the number of suspicious recordings encountered so far. From the *Possible Attack State* the node can move into the *Normal + Knowledge State* when its readings go back to normal; otherwise, it needs to increment its counter value, and check if the number of suspicious readings has crossed the predefined limit. When the counter value becomes greater than the predefined limit, the node moves itself into the *Attacked State* from where it sends out an alert to the base station. In the *Normal + Knowledge State* the node is aware of some abnormalities, but it is still not certain that it is being attacked. It decides to wait until it notices another abnormal observation at which point it moves back into the *Possible Attack State*. The logic behind this approach is that the node tries to eliminate the possibility of false positives by not drawing any conclusions before finding substantial evidence. However, there is no reasonable approach to set the limit value correctly. If it is set too low then the state transition system will most likely be able to prevent the proposed Actuation attack, but at the cost of transmitting too many false positives, which would lead to several of the lWSNs getting cut off from the network. If it is set too high then it is highly unlikely that the legal network will be able to avert an Actuation attack before considerable damage has been done to the network.

# Chapter 8

# Conclusion

Sensor networks are a promising new technology with many important applications, such as environment monitoring, military surveillance, robot control, and health care. Large-scale sensor networks are often deployed in potentially adverse or hostile environments to examine time-dependent physical phenomenon. Due to the unattended operations of the network, an adversary has the opportunity to maliciously actuate the sensor data before it can be recorded by the sensors. In this thesis our research contribution was to establish how randomness and discontinuity may be successfully used by a hostile WSAN to boost the effectiveness and severity of an Actuation attack, which subsequently impairs the decision that a WSN node reports about the absence or presence of a phenomenon to the base station.

The goal of the proposed attack is to prompt the production of needless data packets, called false event reports, which would flood the network, and eventually use up essential network resources, thereby rendering it terminal. The attack can be carried out with the help of a small number of actuator (or actor) nodes that are capable of altering specific environmental conditions around them. These hostile nodes use random self-timers to instigate multiple actuations for varying time spans. As a result of using randomness and discontinuity, different parts of the network are attacked at different time instants, consequently, avoiding detection due to symmetry. For the purpose of our experiments, we used a WSN that recorded temperature inside a room every 31 seconds and used a controlled directional flooding scheme for communicating information to the base station. We designed our attack simulation using 10 hostile actuator nodes that were distributed arbitrarily into the sensing environment. The simulated attack triggered the spawning of numerous false event alerts in the network, which amplified the total number of packets generated and ultimately lead to several dead nodes. Our experimental results suggest that it is possible to boost the severity of the proposed attack by increasing the frequency of actuation, or by increasing the density of hostile actor nodes. We worked on several possible countermeasures for this attack, but none of our techniques could successfully discover or avert the attack. We thereby conclude that randomness and discontinuity can be effectively used by an adversary to build a robust and resilient Actuation attack that is difficult to control or detect.

# Bibliography


[1] A. Czarlinska, D. Kundur, "Distributed Actuation attacks in wireless sensor networks: Implications and countermeasures," *Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, pp. 3–12, April 2006.

[2] A. Czarlinska , D. Kundur, "Towards characterizing the effectiveness of random mobility against Actuation attacks," *Proc. 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS)*, pp. 3-12, April 2006.

[3] A. Czarlinska, W. Luh, D. Kundur, "Attacks on Sensing in Hostile Sensor-Actuator Environments," *Proc. IEEE Globecom*, pp. 1001-1005, November 2007.

[4] J. Newsome, W. Shi, D. Song, A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses," *Third International Symposium on Information Processing in Sensor Networks (IPSN)*, April, 2004.

[5] K. Römer, F. Mattern,"The Design Space of Wireless Sensor Networks," *IEEE Wireless Communications*, Vol. 11, pp. 54-61, December 2004.

[6] I. F. Akyildiz, I. H. Kasimoglu, "Wireless Sensor & Actor Networks: Research Challenges," *Ad Hoc Networks Journal (Elsevier)*, Vol. 2, No. 4, pp. 351-367, October 2004.

[7] A. D. Wood, J. A. Stankovic, "Denial of Service in Sensor Networks," *IEEE Computer*, Vol. 35(10), pp. 54–62, October 2002.

[8] J. Walters, Z. Liang, W. Shi, V. Chaudhary, "Wireless Sensor Network Security: A survey," in *Security in Distributed, Grid, and Pervasive Computing*, Auerbach Publications, CRC Press, April 2007.

[9] H. Kumar, D. Sarma, A. Kar, "Security threats in Wireless Sensor Networks," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 23(6), pp. 39-45, June 2008.

[10] S. Schmidt, H. Krahn, S. Fischer, and D. Watjen, "A Security Architecture for Mobile Wireless Sensor Networks," *1st European Workshop on Security in Ad-Hoc and Sensor Networks*, August 2004.

[11] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An Incremental Self-Deployment algorithm for Mobile Sensor Networks," *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, pp. 113–126, September 2002.

[12] D. R. Raymond, S. F. Midkiff, "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses," *IEEE Pervasive Computing*, Vol. 7, No. 1, pp. 74-81, January 2008.

[13] A. Perrig , J. Stankovic , D. Wagner , C. Rosenblatt, "Security in Wireless Sensor Networks," *Communications of the ACM*, Vol. 47, pp. 53-57, June 2004.

[14] A. Perrig et al., "SPINS: Security Protocols for Sensor Networks," *Wireless Networks 8*, pp. 521-534, December 2002.

[15] T. Roosta, S. Shieh, S. Sastry "Taxonomy of Security Attacks in Sensor Networks and Countermeasures," in *The First IEEE International Conference on System Integration and Reliability Improvements*, IEEE International, pp. 13-15, December 2006.

[16] D. Boyle, T. Newe, "Security Protocols for Use with Wireless Sensor Networks: A Survey of Security Architectures," *Proceedings of the Third International Conference on Wireless and Mobile Communications*, pp. 54, March 2007.

[17] H, Yang, Y. Yuan, S. Lu, W. Arbaugh, "Towards Resilient Security in Wireless Sensor Networks" in *Proceedings. 6th ACM international symposium on Mobile ad hoc networking and computing MobiHoc'05*, pp. 34-45, May 2005.

[18] S. Avancha, J. Undercoffer, A. Joshi, J. Pinkston, "Security for Wireless Sensor Networks," in *Wireless Sensor Networks*, Chapter 12, pp. 253-275, Kluwer Academic Publication, May 2004.

[19] T. Kavitha, D. Sridharan, "Security Vulnerabilities in Wireless Sensor Networks: A Survey," *Journal of Information Assurance and Security*, pp. 31-44, June 2009.

[20] A. S. K. Pathan et al., "Security in Wireless Sensor Networks: Issues and Challenges," *Proceedings of 8th IEEE ICACT 2006*, Vol. 2, pp. 1043-1048, February 2006.

[21] W. Xu, K. Ma, W. Trappe, Y. Zhang, "Jamming Sensor Networks: Attack and Defense Strategies," *IEEE Network*, Vol. 20, No. 3, pp. 41-47, June 2006.

[22] S. Ganeriwal, A. Kansal, M. B. Srivastava "Self Aware Actuation for Fault Repair in Sensor Networks," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Vol.5, pp. 5244 – 5249, May 2004.

[23] F. Hu et al., "Trustworthiness in Wireless Sensor & Actuator Networks: Towards Low-complexity Reliability and Security," *IEEE Global Telecommunications Conference*, December 2005.

[24] A. Czarlinska, D. Kundur, "Coordination and Selfishness in Attacks on Visual Sensor Networks," *Proceedings of IEEE Wireless Communications & Networking Conference (WCNC)*, pp. 2391-2396, April 2008.

[25] S. Meguerdichian et al., "Exposure in Wireless Ad-Hoc Sensor Networks," *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 139-150, July 2001.

[26] A. Tews et al., "Avoiding Detection in a Dynamic Environment," *Proceeding of 2004 IEEE/RSJ international Conference on Intelligent Robots and Systems*, vol. 4, pp. 3773-3778, October 2004.

[27] F. Ye, H. Luo, S. Lu, L. Zhang, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," *IEEE Journal On Selected Areas In Communications*, Vol. 23, No. 4,pp. 839-850, April 2005.

[28] A. Bharathidasan, V. Anand, S. Ponduru, "Sensor Networks: An Overview," *Technical Report,* Department of Computer Science, University of California, Davis, www.csif.cs.ucdavis.edu/~bharathi/sensor/survey.pdf, 2001.

[29] C. Townsend, S. Arms, "Wireless Sensor Networks: Principles and Applications" *Sensor Technology Handbook*, Chapter 22, pp. 575-589, Elsevier Inc., 2005.

[30] I. F. Akyildiz et al., "A Survey on Sensor Networks," *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, August 2002.

[31] Samuel Madden, "Intel Lab Data", db.csail.mit.edu/labdata/labdata.html.

[32] "Gaussian Function", en.wikipedia.org/wiki/Gaussian_function.

[33] "MICA2DOT Wireless Microsensor Mote," www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf.

[34] S. Madden, J. Hellerstein, W. Hong, "Tiny DB: In-Network Query Processing in TinyOS," telegraph.cs.berkeley.edu/tinydb/tinydb.pdf, September 2003.

[35] "TinyOS Documentation Wiki," docs.tinyos.net/index.php/Main_Page.

[36] "The R Project for Statistical Computing," www.r-project.org.