SUCCESS ANALYSIS OF DECEPTION IN WIRELESS SENSOR NETWORKS

By

VANITHA GOPINATH

Bachelor of Science in Electrical and Electronic Engineering

Madurai Kamaraj University

Madurai, Tamilnadu, India

2000

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2010

SUCCESS ANALYSIS OF DECEPTION IN

WIRELESS SENSOR NETWORKS

Thesis Approved:

Dr. Johnson Thomas

Thesis Adviser

Dr. Douglas Heisterkamp

Dr. Debao Chen

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I thank my adviser, Dr. Johnson Thomas, for his patience, and for his incredible guidance throughout, in completing this thesis. I thank and love my husband, Prashanth, for being there for me through all the odds as my support system, and my son, Pranav, for being my motivation.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1

Wireless Sensor Networks – Attacks and Countermeasures 1
Deception in Information Systems 2
Dempster-Shafer Theory for Uncertainty 4
Sinkhole attacks 5

II. REVIEW OF LITERATURE 7

Deception Technologies in Information Protection 7
Dempster-Shafer for combining evidences 8
Sinkhole attacks in WSNs 8

III. METHODOLOGY 9

System Architecture 9
Infrastructure 9
Network Protocol 10
Assumptions 11
Performance Metrics 11
Determining Deception Success 11
Modeling Uncertainty 12
Determining Deception Success with Uncertainty 15
Deception Determination Algorithm 16

Chapter Page

IV. RESULTS 18

Network Setup 18
Simulation 18
Results of Deception Success Analysis 21

V. CONCLUSION 35

REFERENCES 37

LIST OF TABLES

Table Page

1. Attacker Input Modes 20
2. Deception Success Rate for Worst and Best Cases…….26

LIST OF FIGURES

Figure Page

1. Network Topology 10
2. Deception Success Determination 17
3. Worst Case Freq Deviation for Mode 1 21
4. Worst Case Freq Deviation for Mode 2 22
5. Worst Case Freq Deviation for Mode 3 22
6. Worst Case Freq Deviation for Mode 4 23
7. Worst Case Freq Deviation for Mode 5 23
8. Best Case Freq Deviation for Mode 1 24
9. Best Case Freq Deviation for Mode 2 24
10. Best Case Freq Deviation for Mode 3 25
11. Best Case Freq Deviation for Mode 4 25
12. Best Case Freq Deviation for Mode 5 26
13. Hop Count Deviation for Mode 1 27
14. Hop Count Deviation for Mode 2 27
15. Hop Count Deviation for Mode 3 28
16. Hop Count Deviation for Mode 4 28
17. Hop Count Deviation for Mode 5 29
18. Deception Success by Varying Node Reliabilities for Worst Mode 2 30
19. Deception Success by Varying Node Reliabilities for Best Mode 2 30
20. Deception Success by Varying Node Reliabilities for Worst Mode 5 31
21. Deception Success by Varying Node Reliabilities for Best Mode 5 31
22. Attacker Pattern and Deviation in Worst Case Mode 2 32
23. Attacker Pattern and Deviation in Best Case Mode 2 32
24. Attacker Pattern and Deviation in Worst Case Mode 5 33
25. Attacker Pattern and Deviation in Best Case Mode 5 33
26. Deception Success with Best and Worst Predictions 34

CHAPTER I

INTRODUCTION

Wireless sensor networks (WSNs) are a family of wireless networks consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, pollutants, etc, at different locations. Unlike conventional networks, WSNs have some unique constraints, including:

1. Limited computational and storage resources due to miniature form factor and cost restriction;
2. Limited and usually non-replenishable power source;
3. Harsh deployment environment;
4. Dynamic network topology;
5. High hardware and communication failure rate;
6. Unattended operation.

They are highly vulnerable to many types of attacks, including DoS attacks, due to their limited resources. These DoS attacks could be at different sensor network layers. Research has focused on protecting sensor networks using defensive approaches such as key management and encryption techniques. However such techniques are expensive to implement for tiny cheap sensors. Furthermore, given that there may be thousands if not more sensors deployed, encrypting each message is expensive.

Research on responding to an attack or a potential attack has been very limited. Given the limited resources of sensors networks, they are vulnerable to attacks and responding to an attack is difficult. There are a number of possible responses to an attack. For example, the attacker may be ejected from the network. This is almost impossible in a sensor network due to resource constraints. Another response would be to isolate the attacker and re-route traffic so that the attacker is bypassed [14]. The disadvantage with this approach would be that the attacker would change his attack strategy or move to a different part of the network. A third approach would be to increase the protection of the network by implementing encryption or key management. This is of course

resource intensive. An ideal response should place limited burden on resources while maintaining the normal functioning of the network.

DECEPTION IN INFORMATION SYSTEMS

In this thesis we propose deception as a response mechanism. Once suspicious activity or an attack has been detected, deception is applied. The objective of deception is context dependent. For example, another part of the sensor network may be collecting critical data, the objective then becomes to keep the attacker busy and keep him in the same location, thereby buying time.

In our approach, deception is not a resource intensive task as the cost is minimal, and it maintains the normal operation of the network. Deception (also called beguilement) is the act of convincing another to believe the information that is not true, or not the whole truth as in certain types of half-truths. Thus in the case of an attack being sensed by an IDS, deception is employed to mislead the adversary into believing that the attack is successful and to keep the adversary at bay. The deceiving nodes help delay or prevent the attack from propagating further by sending responses typical of a vulnerable victim, thus wasting the attacker's resources.

The two broad categories of deception are dissimulation and simulation. Dissimulation consists of concealing the truth, or in the case of half-truths, concealing parts of the truth, like inconvenient or secret information. There are three dissimulation techniques: camouflage (blend into the background), disguise appearance (altering the model) and dazzle (obfuscate the model). Simulation consists of exhibiting false information. There are three simulation techniques: mimicry (copying another model), fabrication (making up a new model), and attraction (offering an alternative model)

Fowler and Nesbitt [16] suggest six general principles for effective tactical deception in warfare based on their knowledge of air-land warfare. They are:

1. Deception should reinforce enemy expectations.
2. Deception should have realistic timing and duration.
3. Deception should be integrated with operations.
4. Deception should be coordinated with concealment of true intentions.
5. Deception realism should be tailored to needs of the setting.
6. Deception should be imaginative and creative.

Among the several proposed defensive deceptions [10], a few of the best ones are deception in

1. Object, deceiving as to the reality of the files and directories shown;
2. Frequency, deceiving as to the activity rate at a site;
3. Time-at, deceiving as to the timestamps on files;
4. Cause, deceiving as to why files are present;
5. Purpose, deceiving as to what files are being used for;
6. Content, deceiving as to validity of data in files;
7. Measure, deceiving as to file sizes; and
8. Precondition, deceiving as to conditions preventing access to some files and directories.

Apart from using deception for defensive purposes, some researchers use fake honeypots to scare away the attackers [9]. Attackers avoid honeypots for several reasons. First they do not like being monitored, because of their illegal activities. Also they like to save their attack methods for critical moments. They also know they aren't achieving much trying to compromise a honeypot. Also though most honeypots look easy to subvert, they have very rigid controls that prevent the attack from propagating further. The attacker may know there is a better chance of success by avoiding this machine and by starting to attack other machines. In addition, with a fake honeypot, an attacker cannot find any monitoring software to disable, and will be more likely to ignore the honeypot clues on the next real honeypot they find [12]. Fake honeypots could be effective defenses if used occasionally on critical systems. But even if fake honeypots were widespread and ineffective, the effectiveness of true honeypots would increase in compensation.

With hackers being well aware of the deceptive mode of defense, they look for traits in the systems that they attack for possible signs of deception, like presence of unconscious clues (short responses, vagueness, lack of digital information, unexpected processing delays, and clues from real-time behavior) and inconsistencies (due to inadvertent mistakes made by the deceiver). In the process of evolution of honeypots, we see that focus is on improvising the deception by constant vigilance and coherent responses to the actions of the attacker. Thus it becomes necessary to analyze the attacker responses to make sure the deception is successful. If the attacker backs away from attacking the deceiving node further, it might mean that the attacker has found a more desirable victim or that the attacker has become wary of the deceiving node. This causes the deception to fail. That is why deception success is such an important metric for a network under attack.

DEMPSTER-SHAFER THEORY FOR UNCERTAINTY

In addition to applying deception, the success of the deception must also be measured. If the deception is not successful, the defending sensor network has to change its approach. In this paper, we focus on measuring the success of deception using the Dempster-Shafer theory. Dempster-Shafer theory handles uncertainty [11]. The uncertainty in this thesis focuses on the intentions of the attacker and the reports on deception success provided by the deceiving nodes. The uncertainty of the attacker's intentions is updated according to the progressive behavior of the attacker. We use the Dempster-Shafer theory to consolidate the results obtained from various deceiving nodes. The Dempster-Shafer theory, also known as the theory of belief functions, is a generalization of the Bayesian theory of subjective probability. Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question on probabilities for a related question. The Dempster-Shafer theory is based on two ideas: the idea of obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule for combining such degrees of belief when they are based on independent items of evidence.

Without the need for a priori knowledge of the conditional probabilities as in other approaches such as Bayesian, the Dempster-Shafer theory enables combining metrics on deception success which are obtained from various deceiving nodes. We work with the scenario where the network is under an attack; studying the request-response pattern for that particular attack; exhibiting vulnerabilities to lure the intruder into channeling its resources on launching attack on the deceiving nodes; sending responses that suggest that the attack is working efficiently; and analyzing the success of deception based on the responses from the attacker in turn.

We measure deception success for a particular attack by identifying the pattern of the attacker's network traffic. The requests from attacker are responded to in a desirable way to lead the attacker to believe that the deceiving node is being compromised. The expected response from the attacker is compared to the actual response for a given attribute of the attacker's response. If the response deviation along with the uncertainty exceeds a threshold value for that attribute, then it means that the deception has failed. The success of the deception process is in keeping the response deviation along with uncertainty below the preset threshold value. Another fact to consider is that with less deviation of the actual from the expected response, uncertainty keeps decreasing.

SINKHOLE ATTACKS

The proposed approach will be validated by studying the sinkhole attack which is a DoS attack. In a sinkhole attack, the adversary's goal is to lure nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center. Because nodes on, or near, the path that packets follow have many opportunities to tamper with application data, sinkhole attacks can enable many other attacks (selective forwarding, modifying or even dropping the packets coming through).

Sinkhole attacks typically work by making a compromised node look especially attractive to the surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high-quality route to the base station. Some protocols might actually try to verify the quality of the route with end-to-end acknowledgements containing reliability or latency information. In this case, a laptop-class adversary with a powerful transmitter can actually provide a high-quality route by transmitting with enough power to reach the base station in a single hop, or by using a wormhole attack. Due to either the real or imagined high-quality route through the compromised node, it is likely that each neighboring node of the adversary will forward packets destined for a base station through the adversary, and also propagate the attractiveness of the route to its neighbors. Effectively, the adversary creates a large ''sphere of influence'', attracting all the traffic destined for the base station, from nodes that are several hops away from the compromised node.

According to the multihop routing algorithm, [4] identifies three ways for an attacker to launch the sinkhole attack.

1. Advertise a low path cost with its parent
2. Make other nodes look like they have worse path costs than itself
3. Change its parent to the neighbor with the minimum path cost

We focus on the first option of attack. The normal sensor nodes eavesdrop on the neighboring nodes' traffic (ping), and calculate the cost based on the number of messages that the neighboring node sends (the more messages heard from a node, the higher the quality of its link is). The sinkhole attacker broadcasts the routing messages at a higher frequency, so that a normal node will be hearing many more messages from the attacker than from its other neighboring nodes. And with a very low hop count advertised, the attacker could always claim that it is closer to the base station than its parent. This triggers a parent changing mechanism at the node that is eavesdropping and it chooses the attacker as the next hop based on the cost calculated and the lowest hop counts to the base station.

The two parameters that we will be looking at are:

1. The frequency of advertisements of the attacker and
2. The quality of the route advertised. (essentially the hop counts from the base station)

The deception analysis algorithm uses this data collected, compares these parameters to the predictions and analyses the deception success.

Section 2 of the thesis covers the history of related work. Section 3 covers the problem at hand, namely analysis of the deception success, proposed architecture, methodology, performance metrics, and incorporating uncertainty. Section 4 shows the results obtained by the work done in the thesis period and section 5 gives the concluding remarks.

CHAPTER II

REVIEW OF LITERATURE

DECEPTION TECHNOLOGIES IN INFORMATION PROTECTION

Neil C. Rowe works on deception using the fake file systems and studies the processes of counter-deception against that and also the means of successful deception using counter-counter-deception [1]. The approach defines consistency as compatibility with average statistics. Comparison of vectors between fake and real systems is used as a guide to design the fake. It talks about the most elaborate tool for evaluating deception, in particular a metric-calculating tool that summarizes a file system by a vector of 72 numbers and comparing their values to the respective values of a honeypot. A mathematical model for deception is proposed in [15].

Usually attackers look for obvious clues like unconscious clues and inconsistencies. Low-interaction deception could be detected by launching different service requests and examining an emulated service. Furthermore, transmission features, such as latency, error, or protocol header could be collected and an analysis of the connection features could be used to detect a fabricated operating system or a virtual network interface. System level features like types of physical devices, types of file systems, memory usage of hidden programs, etc., can be used to detect any high-interaction honeypot, deployed at a physical or virtual machine. Thus the detection in general, could be service support detection, connection feature detection, or system level detection [13].

Some ordinary systems plant obvious clues that suggest a honeypot, as a form of 'vaccination' to scare away the attackers [9]. There is a study on such fake honeypots that examines the possible clues to a honeypot that an attacker looks for. It experiments the response of an attacker to a fake honeypot as well as a real honeypot. They are useful only if used occasionally in critical systems. There is use of 'fake fake honeypots' where real honeypots pretend to be obviously fake honeypots. There is also discussion of the evolution of honeypots and fake honeypots in information warfare.

Honeyanole is a deception system proposed to successfully escape honeypot hunting, while collecting attack information using a 3-phase process of collection, redirection and deception [2]. The collection module builds an orderly blacklist of possible attackers; the redirection module channels the incoming traffic to a production server or deception server with the aid of the blacklist; the deception module captures the intrusive processes. Since low-interaction honeypot could be discovered by service support detection, and high-interaction honeypot could be discovered by system level detections, a new technique called medium-interaction honeypot is suggested for better deception technology.

The most notable application of deception in modern networked information system is the Honeynet, which is a high-interaction honeypot, a network of real computers for attackers to interact with and is designed to capture extensive information on threats [21]. Some research shows remarkable results for conventional information systems [5][6] [20]. But these theories developed do not translate to sensor networks, due to differences between sensor networks and conventional systems.

DEMPSTER-SHAFER FOR COMBINING EVIDENCES

The Dempster-shafer theory is used in the context of combining data from multiple nodes to estimate the likelihood of intrusion in a wireless ad hoc network [11]. The Dempster-Shafer theory is especially suited in such cases where the observing nodes may not be reliable. It offers a mathematical way to combine evidence from multiple observers without the need to know about a priori or conditional probabilities as in the Bayesian approach.

Dempster-Shafer and Bayes approaches are compared in [17]. While probability theory takes it as given that something either is or isn't true, Dempster-Shafer theory allows for more nebulous states of a system (or really, our knowledge), such as "unknown". It can of course sometimes be far safer to be undecided about what a target is, than to decide wrongly and act accordingly with what might be disastrous consequences, as in battlefield surveillance. Bayes theory requires prior probabilities, and DS requires some preliminary assignment of masses that reflects our initial knowledge of the system. The computational disadvantages attribute towards the lack of DS popularity. One advantage of using one approach over the other is the extent to which prior information is available.

Braun [7] shows that both methods are robust over the entire sensor information domain, and generally where one succeeds or fails the other will do the same, with just a slight edge being given to Dempster-Shafer as compared with the Bayes approach. Dempster-Shafer is used as a multi sensor fusion method in [8].

SINKHOLE ATTACKS IN WSNS

The strategies that an attacker can follow to successfully launch a sinkhole attack is clearly stated in [4]. It also states the specific detection rules that the legitimate nodes need to follow in order to detect an ongoing attack.

Although some work has been done on deception improvisation, no-one has come up with a scheme to measure the success of deception itself. Our deception analysis system evaluates the success of the deception process, along with considering the uncertainty in picture, using the

Dempster-Shafer theory. Deception analysis is essential to provide better and a more improvised form of deception in this arms race of attack and deception.

CHAPTER III

METHODOLOGY

In the deception scenario considered, the adversary's network traffic is studied and the responses are sent back according to the pattern observed. In our proposed work, the deceiving nodes study the response from the attacker and calculate the success of deception based on the difference between the expected response and the actual response. Each deceiving node calculates success or rather the discrepancy of the actual from the expectation. The observations made by each of the deceiving nodes are consolidated and the overall success of the deception process is measured. The uncertainty of deception's success in the initial phase of deception is handled using the proximity of the attacker responses to the predictions. Dempster-Shafer is used for the consolidation of the deception success metrics received from various deceiving nodes employed in the attacker's vicinity.

SYSTEM ARCHITECTURE

We focus on the system architecture from the security and performance perspective, since our work has to do more with tackling the attacks, deception and its success.

INFRASTRUCTURE:

The infrastructure includes a base station, cluster heads, sensors and their current deployment status. A wireless sensor network is characterized by the limited power they can harvest or store, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, communication failures, heterogeneity of nodes, large scale of deployment and unattended operation. Each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station). Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth.

In our deception technology, we adopt a distributed approach to eliminate the delay caused by long trips of packets. In this distributed network topology, the entire sensor field is divided into subzones. Each subzone has at least one Distributed Processing Unit (DPU), which have computational and storage capacities stronger than the regular sensor nodes.

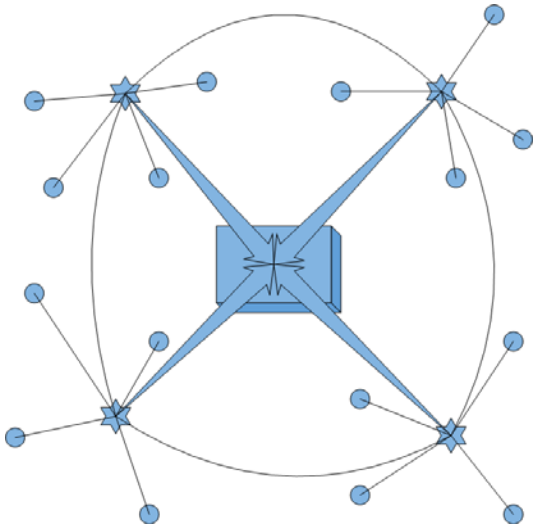

FIGURE 1: NETWORK TOPOLOGY

The DPUs take care of local data processing and decision making in its subzone. Code related to deception is stored only in DPUs. Although it makes the network topology complex and more hierarchical, the DPUs make up for that by enabling faster fabricated responses thus making deception more believable. When an attack is detected or suspected by the Intrusion Detection System, the nodes selected to carry out deception will receive installation of code for deception functionalities from a nearby DPU. As DPUs are generally considered more secure than individual sensor nodes, the deception code are protected from tampering and the defender's intention of deception is well hidden from the attacker.

The concept of sub-zone further helps us enable better power management over each zone. For sub-zones which have low activities or low risks, both DPUs and observing functions on nodes can be shut off temporarily.

NETWORK PROTOCOL:

The proposed framework employs a group key management scheme. A group key enables the base station to communicate with the entire network. An individual key also enables the base station to communicate individually with each node. If an attacker is detected, the base station uses the individual keys to reset the group key so that the attacker (assuming it has access to the previous group key – it may be a sensor node) is excluded from the group. Hence, the rest of the network can communicate normally without the attacker being able to read the messages. The base station informs the selected nodes using their individual keys to act as deceiving nodes and sends

them the different parameters. Then the base station informs the nodes that are in the vicinity of the deceiving nodes not to direct any traffic to these nodes.

ASSUMPTIONS:

Generally, if an entity can be deceived, it must have abilities to evaluate its surrounding environment and make decisions. Our first assumption is that the attacker whom the defender tries to deceive should have the abilities to monitor the victim network's reactions to the attack; furthermore, it should have the intention to maximize the outcome of the attack. Adversaries are cost restricted, which means that, economically, adversaries have the intention to minimize the cost of attacks while achieving the attacking goals.

For different types of attacks, the first assumption may not be always valid as retrieving feedback from the network may not be possible in some kind of attacks despite the attacker's interest to do so. This means this deception framework is not applicable to all kind of attacks. We urge that the second assumption is valid as (1) equipment and personnel are limited even in a state-backed, military based attack; (2) excessive attack may lead to exposure of the attacker.

We also assume that the base station has an overview of the whole network and a distributed intrusion detection system (IDS) is available that can identify the type of attack and the IDS reporting the attack will give an indication of the region of origin of attack. We assume the IDS report is available to the base station. The base station is responsible for formulating an appropriate response to the intrusion report provided by the IDS.

PERFORMANCE METRICS

The calculations on deception success are made individually by each of the deceiving nodes. The deceiving nodes study traffic patterns by observing the attacker and recording the parameters [3]. We assume an attacker sends requests and a deceiver responds to these requests. A function *RespToReq*(*ReqToResp*(*x*)) predicts the expected requests from the attacker. Each deceiving node executes this function to predict the behavior of the attacker. The attacker sends request *Req*(*x*). Given the request from the attacker the deceiver responds as *ReqToResp*(*x*). Given the response by the deceiving node, the deceiving node predicts the future requests of the attacker to be *RespToReq*(*ReqToResp*(*x*)).

The output of this function for our purposes is a tuple ($a_{predicted}$, $b_{predicted}$) where $a_{predicted}$ and $b_{predicted}$ are the two attributes that are being observed from the attacker's response. We assume that this is the expected requests from the attacker over the next time period. In other words, the values that the deceiving node expects in the response from attacker for different attributes are given as an objective tuple ($a_{predicted}$, $b_{predicted}$).

DETERMINING DECEPTION SUCCESS:

The difference between the expected values for the attributes and the actual is calculated and difference is expressed as a tuple

$$d(a,b) = (|a_{predicted} - a_{actual}|, |b_{predicted} - b_{actual}|) \quad (1)$$

Priority may be given to different attributes. The weighted difference is therefore represented as

$$d(w_1 a, w_2 b) = (w_1|a_{predicted} - a_{actual}|, w_2|b_{predicted} - b_{actual}|) \quad (2)$$

There are the respective thresholds A_T and B_T, the deviation values below which are acceptable. Beyond these thresholds, the deception is not working. That is, as long as

$d(w_1 a, w_2 b) < (A_T, B_T)$, the deception is working.

We use a pairwise comparison, that is,

$$d(w_1 a, w_2 b) < (A_T, B_T) \text{ iff } w_1|a_{predicted} - a_{actual}| < A_T \text{ and } w_2|b_{predicted} - b_{actual}| < B_T \quad (3)$$

MODELING UNCERTAINTY:

There is however uncertainty involved. There are 2 factors in particular which contribute to uncertainty:

- Uncertainty in the attacker's intentions
- Uncertainty in the requests received by the deceiver. Although the deceiving nodes are assumed to be trustworthy, they may be low on battery or they may be far from the attacker which means the values reported by the deceiving nodes may not be completely reliable and may contain errors.

We use the Dempster Shafer theory (DST) [11] to capture the belief and uncertainty. Belief tells us how certain we are that the deception is succeeding. The uncertainty introduces the concept of plausibility from DST. Plausibility tells us if the uncertainty was solved in favor of our hypothesis. For example, we are sure that the deception is working with a probability of say *x.* However beyond that, there is uncertainty.

We formalize the deception problem in terms of a universe of mutually exhaustive distinct hypotheses *H*. In our case the set *H* is {*DS,DF*} where *DS* is 'deception succeeding' and *DF* is 'deception failing'. We define a *bpa* (basic probability assignment) as a function *m* that maps subsets of *H* to [0,1], such that *m*(*X*) is between 0 and 1 for all subsets *X* of *H*, and

$$\sum_{X \subseteq H} m(X) = 1 \quad (4)$$

The Belief function, from subsets of *H* to [0, 1] gives a measure of the belief of a subset of the hypothesis.

$$Bel(A) = \sum_{X \subseteq A} m(X) \tag{5}$$

Bel(H)= 1, and *Bel(ø)=0,* where ø is the empty set.

The plausibility function *Pl* which captures the uncertainty is

$$Pl(A) = 1 - Bel(\neg A) \tag{6}$$

where ¬*A* is the set complement of *A* in *H*, i.e., *H* – *A*.

Example

Based on the evidence seen so far there is a belief that the deception is succeeding with a 0.8 probability. There is an uncertainty of 20 percent – the deception may be working or it may not be working. Therefore the belief is 0.8 and the plausibility is 1, that is, [0.8,1.0]

Let us assume 2 deceiving nodes:

N1 - is reliable with .8 probability. In other words, if N1 says that deception is working there is a .8 probability, that it is true.

N2 - is reliable with .7 probability. In other words, if N2 says that deception is failing there is a .7 probability, that it is true.

Therefore *H = {DS, DF}*

For N1: *m1({DS})* = 0.8

In other words, this is the belief that

m1(H) = m1({DS, DF}) = 0.2 (this is the uncertainty)

For N2: *m2({DF})* = 0.7

m2(H) = m2({DS, DF}) = 0.3 (this is the uncertainty)

The DST for combining evidence is:

$$m3(A) = \frac{\sum_{X \cap Y = A} m1(X) * m2(Y)}{1 - \sum_{X \cap Y = \phi} m1(X) * m2(Y)} \tag{7}$$

Let us suppose that both N1 and N2 say that from evidence the deception is working.

Therefore we want to find out *m3(DS)*

The probability that both of them are providing unreliable information is .2*.3 = .06

Therefore there is a 94% chance that at least one of them is giving the right information, which is, the deception is working.

All the evidence is consistent, and as long as the two pieces of evidence are independent, the conclusion is justified. Therefore the belief is 0.94 and the plausibility is 1, that is, [0.94,1.0]

Now suppose N1 says that from what it can see in the evidence is the deception is working, but N2 says the evidence is that deception is not working.

m1(DS) = 0.8

m2(DF) = 0.7

Since there is no evidence from N2 about the deception being a success, *m2(DS)* = 0

From (4), *m2(DS) + m2(DF) + m2(DS,DF) = 1*

Thus, *m2(DS,DF)* = 0.3

Therefore:

$$\sum_{X \cap Y = A} m1(X) * m2(Y) = \sum_{X \cap Y = DS} m1(DS) * m2(DS, DF) = 0.8 * 0.3 = 0.24$$

$$1 - \sum_{X \cap Y = \phi} m1(X) * m2(Y) = 1 - (m1(DS) * m2(DF)) = 1 - (0.8 * 0.7) = 0.44$$

Therefore: *m3(DS)* = 0.24/0.44 = 0.55

Now to obtain *m3(DF)*, that is as reported by N2

Therefore:

$$\sum_{X \cap Y = A} m1(X) * m2(Y) = \sum_{X \cap Y = DF} m1(DS, DF) * m2(DF) = 0.2 * 0.7 = 0.14$$

$$1 - \sum_{X \cap Y = \phi} m1(X) * m2(Y) = 1 - (m1(DS) * m2(DF)) = 1 - (0.8 * 0.7) = 0.44$$

Therefore: *m3(DF)* = 0.14/0.44 = 0.32

Thus, *Bel(DS)* = 0.55

From (3), *Pl(A) = 1 – Bel(¬A) = Pl(DS) = 1 – Bel(¬DS) = 1 – Bel(DF)* =1 – 0.32 = 0.68

Therefore belief the deception is succeeding is 0.55 and the plausibility is 0.68, that is, [0.55,0.68]. The degree of uncertainty is therefore 0.68-0.55 = 0.13

DETERMINING DECEPTION SUCCESS WITH UNCERTAINTY:

In one of the previous sections, we saw that if $d(w_1a, w_2b) < (A_T, B_T)$, the deception is working. We assume the weights $w_1 = w_2 = 1$.

Since there is uncertainty involved, both *a* and *b* may be below their thresholds A_T and B_T respectively, but the sum of the belief along with the uncertainty factor may be above the threshold. We assume that initially the uncertainty is in our favor, that is, the deception is successful. However, over time the uncertainty value is deemed to be more unfavorable.

If $a > A_T$ or $b > B_T$, then deception is not working. Let the uncertainty factors be u_a and u_b for the attributes *a* and *b* respectively. To determine the deception success:

function **dec_det**

while *a* and *b* lies within a range **then** {

if $a + f(u_a) > A_T$ **then** deception failed

if $b + f(u_b) > B_T$ **then** deception failed

calculate next values of *a,* u_a and *b,* u_b

}

The functions $f(u_a)$ and $f(u_b)$ are used to recalculate the uncertainty values based on the discrepancies *a* and *b* respectively.

But this is the individual determination of each of the deceiving node. It is necessary to consolidate these readings from different deceiving nodes, which is where Dempster-shafer comes into picture.

Given the deception success determination for two deceiving nodes, we would construct the combined belief as given in equation (7).

The DST for combining evidence is:

$$m3(A) = \frac{\sum_{X \cap Y = A} m1(X) * m2(Y)}{1 - \sum_{X \cap Y = \phi} m1(X) * m2(Y)}$$

DECEPTION DETERMINATION ALGORITHM

1. Determine the expected value for each of the observed attributes for the next time period.
2. On receiving the actual request from the attacker, calculate the deviation of the actual parameter value from that of the expected value for each of the attributes.
3. Recalculate the uncertainty value associated with the attacker's intent using the deviation value from step 2.
4. Determine if the net deviation value (including uncertainty) is below the preset threshold value for each of the attributes.
5. If any of the attribute deviation value exceeds threshold, then the deception has failed for that time period. If not, the deception is a success for that time period.
6. The forged response is sent to the attacker.
7. Repeat steps 1 through 6 for the following time period.

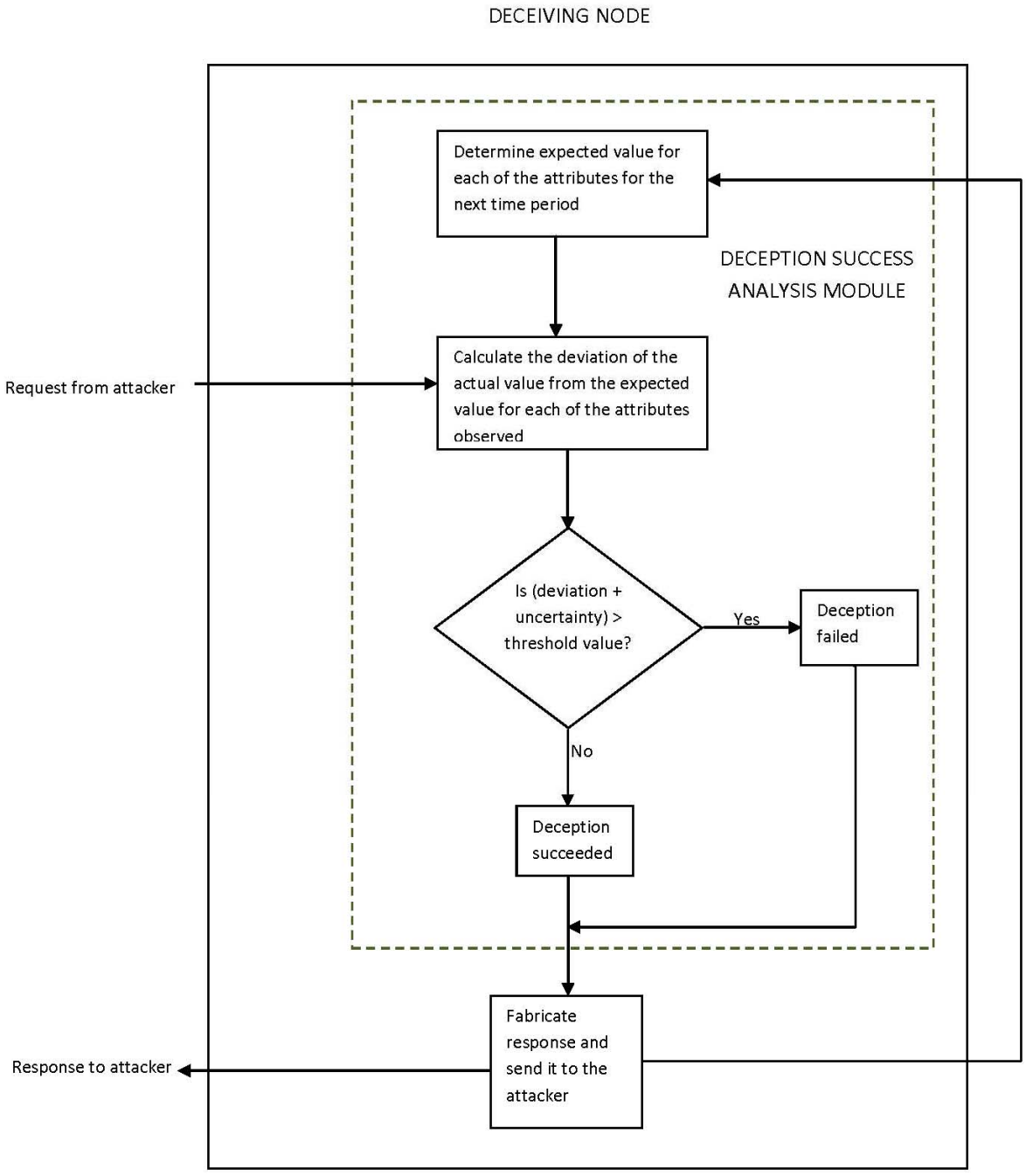

FIGURE 2: DECEPTION SUCCESS DETERMINATION

CHAPTER IV

RESULTS

NETWORK SETUP

The network model works with the normal operation being the data sent to the base station. In the event of an attack being detected, the adversary attack is studied to learn traffic pattern, and appropriate responses are generated and sent to the attacker. This deceiving traffic is to keep the attack from propagating further to other nodes. The sacrificial and deceiving nodes spoof their addresses so that the attacker would think it is getting packets from sensors that are more than one hop away and the sacrificial nodes also spoof the address of the attacker's neighboring nodes that are not sacrificial nodes.

SIMULATION

Our model simulates the deception behavior of a sub-zone as an entity. This entity has deceiving nodes that are employed in the sub-zone apart from the regular sensor nodes of the network. These deceiving nodes individually measure the deception success and report to the DPU (Distributed Processing Unit) of that sub-zone. The DPU collects these reports and makes an informed decision as to if the deception was a success or not for that time period. These calculations are ongoing as the deception progresses.

The type of attack considered for the case study is the sinkhole attack. Predictions are made for each fixed time period based on what is expected from an attacker in that situation and the attacker behavior in previous time periods. There are two parameters that the attacker traffic is studied for – frequency of the sensing messages and also the quality of link advertised by the attacker (in terms of hop counts from base station). Predictions are made based on the typical behavior of the attacker and also based on the attacker’s behavior in the previous time points. These predictions are compared to the actual parameter values obtained from the attacker’s behavior.

Sinkhole attack simulation: The actual attacker parameters are obtained from simulation of a sinkhole attack using the TOSSIM simulator [18]. The attacker data is obtained in the form of a dataset consisting of messages in the following packet format:

[Msg type : Source Node, Original Node, Receiving Node, Local Time of Source node, Hop counts to Base Station];

where

- Msg type:
  S: Sensing Message
  R: Routing Message
- Source Node: The node that is sending the message.
- Original Node: The node that first creates the message and sends it.
- Receiving Node is the node that source node sends the message to.
- Local time of the source node - The simulated system is not perfectly time-synchronized, so only the local time is recorded.
- Hop counts to Base Station - The attacker selects hop counts to the base station randomly between 0 - 2.

There are a total of 25 nodes in the simulation: 1 attacker (Id=1), 1 Base Station (Id=0), 23 Normal nodes (Id=2-24).

Attacker Scheme:

- Link Quality: Claim highest link quality when sending routing message.
- Frequency: Attacker changes the frequency of sending sensing message. (Table 1)
- Rate of sending routing message is the same with normal nodes.

Several different modes of attacker input were used with regard to the frequency of sensing messages to test the deception analysis system, as given in the Table 1. Each attacker mode was obtained from a different dataset from the simulation.

The attacker parameter values under consideration were predicted using a prediction algorithm that considers the behavior of the attacker at previous time points. The attacker traffic from the simulation is studied and used in the prediction algorithm [19] to predict attacker traffic for future time points. The predictions made can be described as “x step ahead look back after y”. Here “x step ahead” means the “x” number of predictions made for future time points per execution of the prediction algorithm. Also “look back after y” means attacker’s historical data is scanned again after every “y” predictions. The predictions are made and there are two cases considered here – the worst and the best. The predictions are better when the history of attacker traffic is referenced more often and the number of predictions made at a time point is less, and for worse cases, it is vice

versa. In our case, we consider the worst case to be “10 steps ahead look back after 50” and best case to be “1 step ahead look back after 10”.

There is uncertainty in the case scenario considered, in respect to the attacker’s intention. The deviation of actual from the predicted, adds to the uncertainty of the attacker’s intentions. The closer the actual parameter values are to the predictions, the more certain are the attacker’s intentions. The uncertainty for any time period is calculated as below:

Let us define α, where $0 \leq \alpha \leq 1$

μ_{Total} - Total value of uncertainty for any time period

μ_n - Current value of uncertainty

μ_{n-1} – Value of uncertainty for previous time period etc.

Each individual uncertainty is calculated from the deviation of actual parameter value from that of the predicted for that given time period. The total uncertainty is based on the current uncertainty and the uncertainties for the previous five time periods. Thus the total uncertainty for a given time period ‘n’ is given as

$$\mu_{Total} = \alpha\mu_n + (1-\alpha)\alpha\mu_{n-1} + (1-\alpha)^2\,\alpha\mu_{n-2} + (1-\alpha)^3\,\alpha\mu_{n-3} + (1-\alpha)^4\,\alpha\mu_{n-4} + (1-\alpha)^5\,\alpha\mu_{n-5}$$

The net deviation is the uncertainty added to the difference between the predicted and the actual values.

There is also uncertainty with regard to the fact that the deceiving nodes have resource constraints too and also the possible distance of the attacker from the deceiving nodes, making the requests received by the deceiving nodes from the attacker unreliable. Here we use the Dempster-Shafer theory at the DPU level to consolidate the reports from the deceiving nodes, incorporating this uncertainty.

Attacker input mode	Description
Mode 1	Transmission frequency increasing linearly
Mode 2	Transmission frequency increasing randomly
Mode 3	Transmission frequency decreasing linearly
Mode 4	Transmission frequency decreasing randomly
Mode 5	Transmission frequency combined sine wave

TABLE 1: ATTACKER INPUT MODES

RESULTS OF DECEPTION ANALYSIS

The success of the deception itself is determined by whether the net deviation for a particular time period stays within the threshold value for that particular time period. The threshold value for any particular time period is based on the parameter value predicted for that time period. We have studied the deception analysis with frequency thresholds varying at 10%, 20% and 30% of the predicted frequency for that time period.

The graphs below show the net deviation of the frequency of sensing messages against the frequency threshold for the worst cases. The time points where the deviation exceeds the threshold values are where the deception failed, according to the deceiving node making the observations. Varying thresholds that are 10%, 20% or 30% of the predicted frequency are shown.

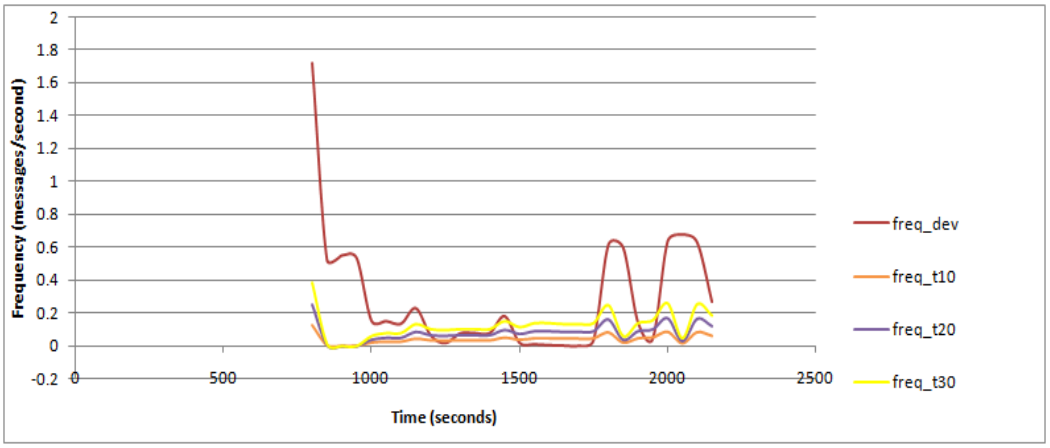

FIGURE 3: WORST CASE FREQ DEVIATION FOR MODE 1

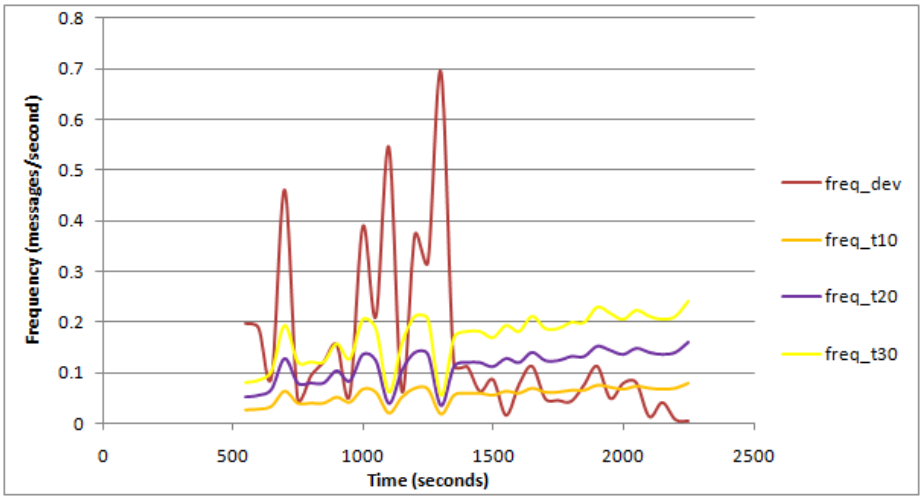

FIGURE 4: WORST CASE FREQ DEVIATION FOR MODE 2

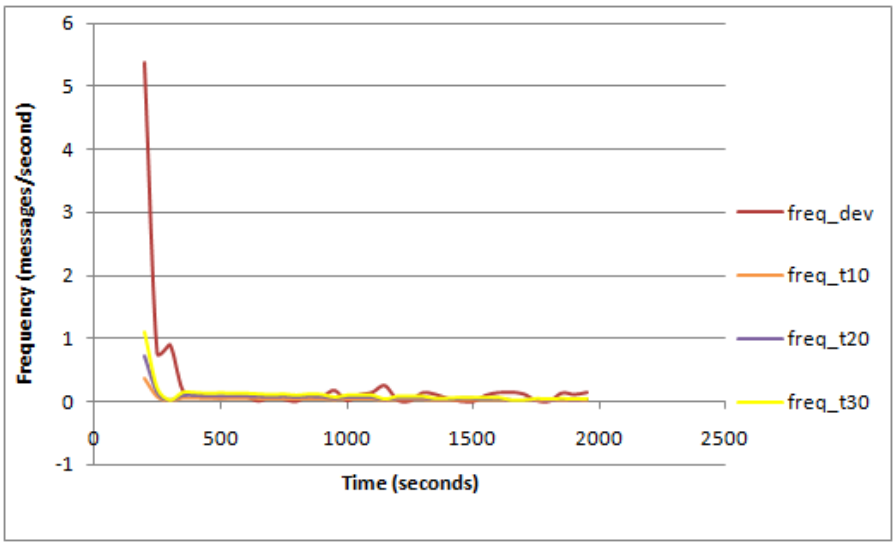

FIGURE 5: WORST CASE FREQ DEVIATION FOR MODE 3

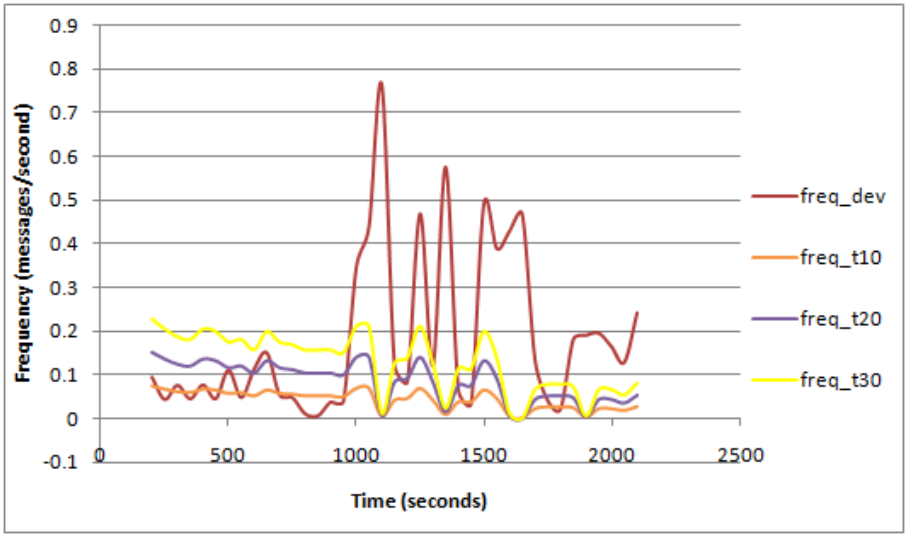

FIGURE 6: WORST CASE FREQ DEVIATION FOR MODE 4

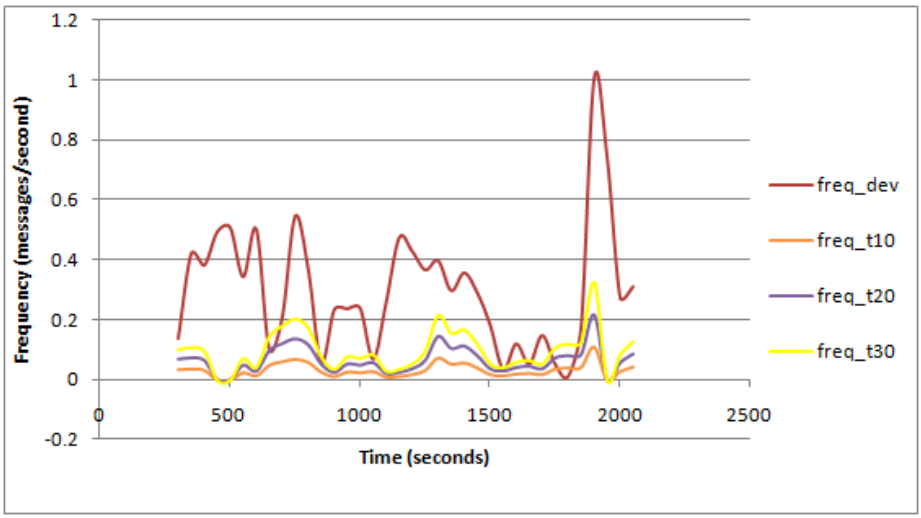

FIGURE 7: WORST CASE FREQ DEVIATION FOR MODE 5

The graphs below show the net deviation of the frequency of sensing messages against the frequency threshold for the best cases. The time points where the deviation exceeds the threshold values are where the deception failed, according to the deceiving node making the observations.

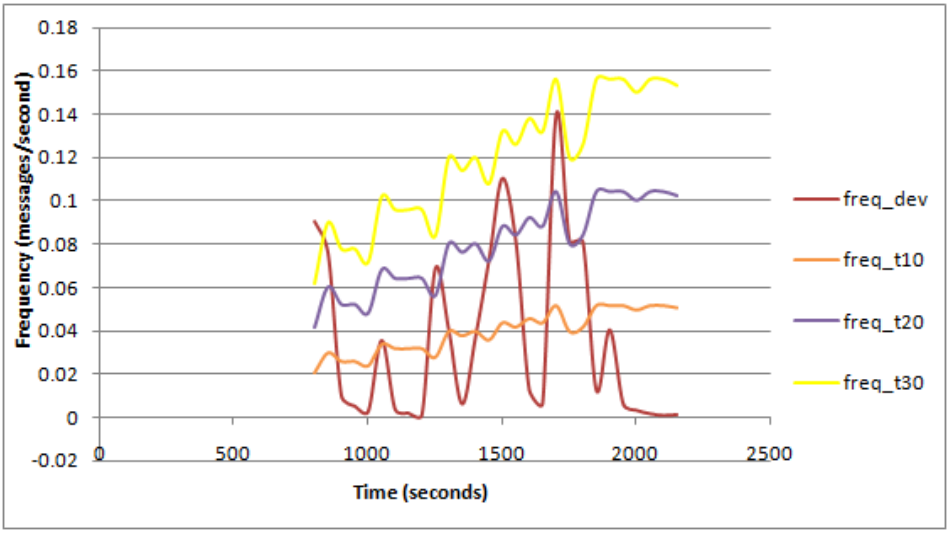

FIGURE 8: BEST CASE FREQ DEVIATION FOR MODE 1

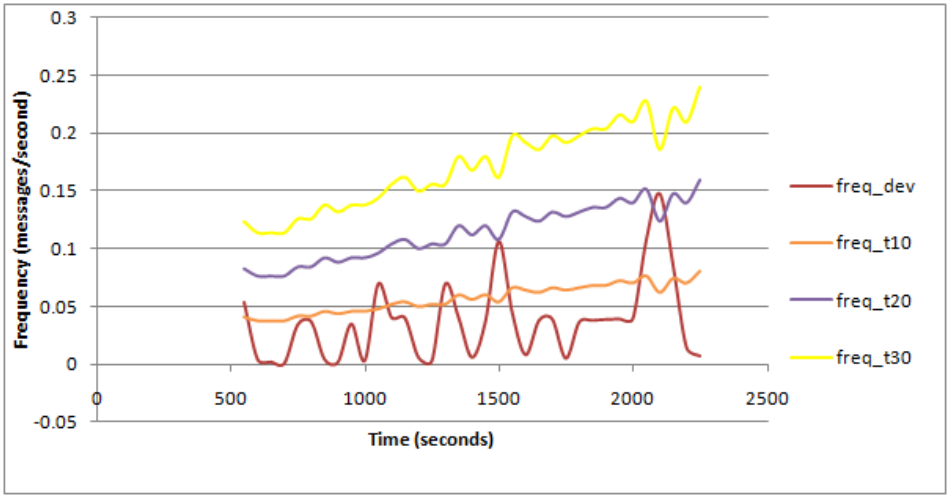

FIGURE 9: BEST CASE FREQ DEVIATION FOR MODE 2

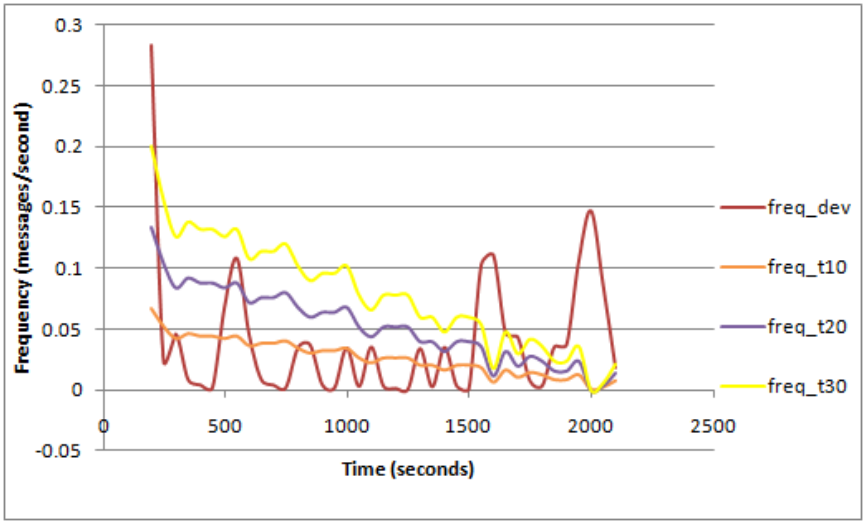

FIGURE 10: BEST CASE FREQ DEVIATION FOR MODE 3

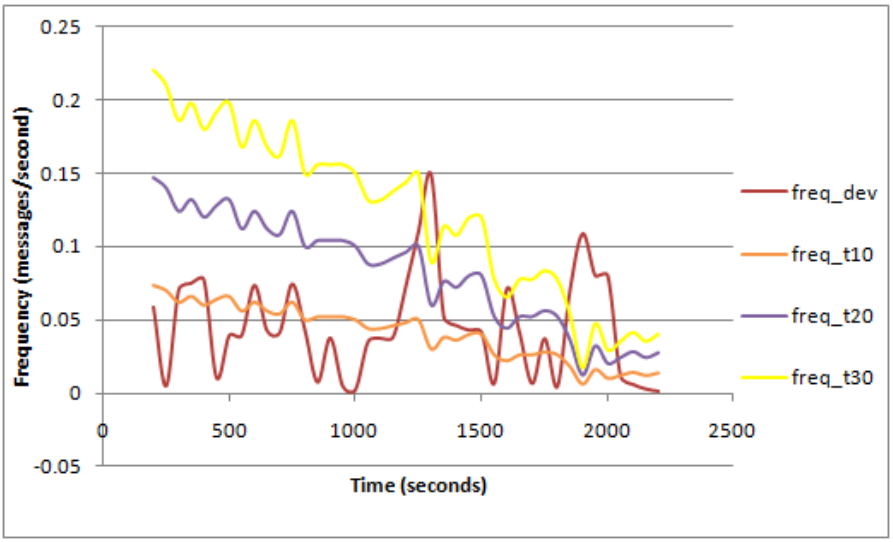

FIGURE 11: BEST CASE FREQ DEVIATION FOR MODE 4

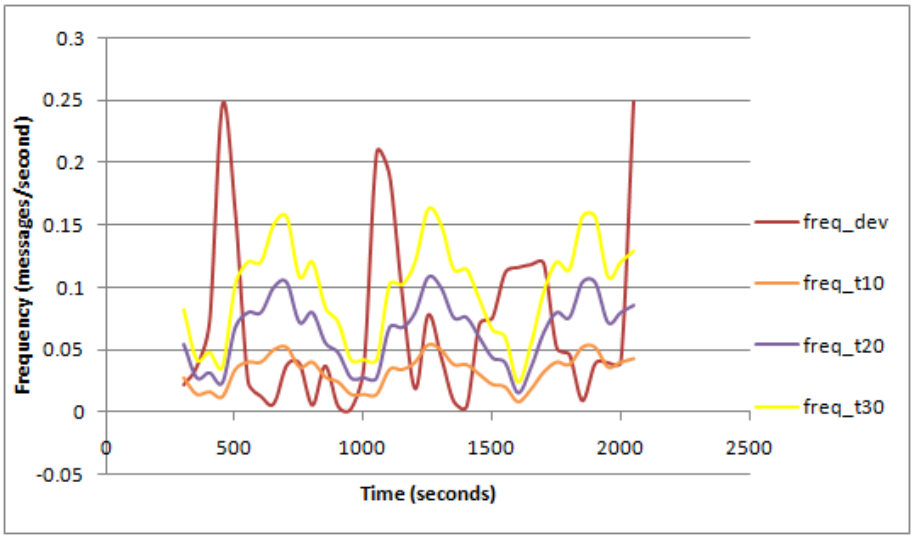

FIGURE 12: BEST CASE FREQ DEVIATION FOR MODE 5

From the graphs we see that the number of times the threshold is crossed almost double or triple with worse cases considered, when compared to the best case scenario for the respective modes as shown in the table below.

Attacker input	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5
Worst	19	14	24	20	34
Best	7	1	13	7	15

TABLE 2: DECEPTION SUCCESS RATE FOR WORST AND BEST CASES

In some cases, as in mode 2, there is a very big difference in the threshold crossing count. This may be attributed to the fact that the mode 2 has attacker input increasing randomly. During the initial pattern learning phase of the prediction, it takes some time for the predictions to stabilize in the worst case scenario, since the look back for worst case predictions is less frequent compared to the best case predictions.

An obvious point from the graphs above for frequency is that the higher the threshold value is set, the more the number of deviation points that fall below the threshold. For example, with worst mode 5, the number of time points at which deviation exceeds threshold (i.e., deception fails) is 35 when the threshold is 10% of prediction, 34 when the threshold is 20% of prediction and 29 when the threshold is 30% of prediction. Setting threshold to accommodate deviation can be made as lenient or as strict as the need may be.

The graphs below show the net deviation of the hop count advertised by the attacker against the hop count threshold. The time points where the deviation exceeds the threshold values are where the deception failed, according to the deceiving node making the observations.

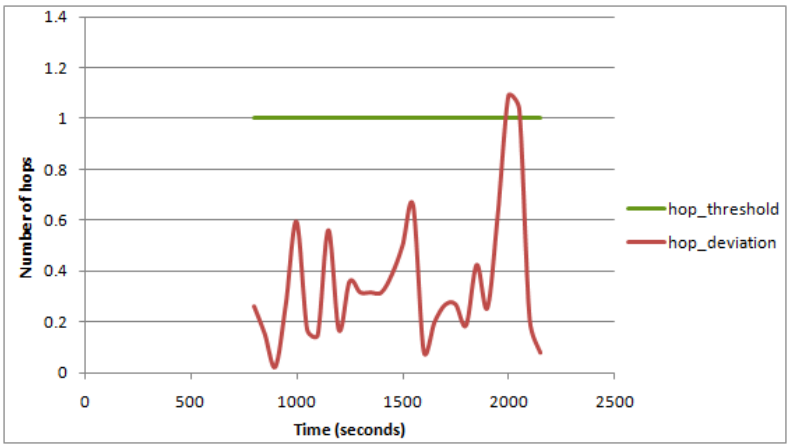

FIGURE 13: HOP COUNT DEVIATION FOR MODE 1

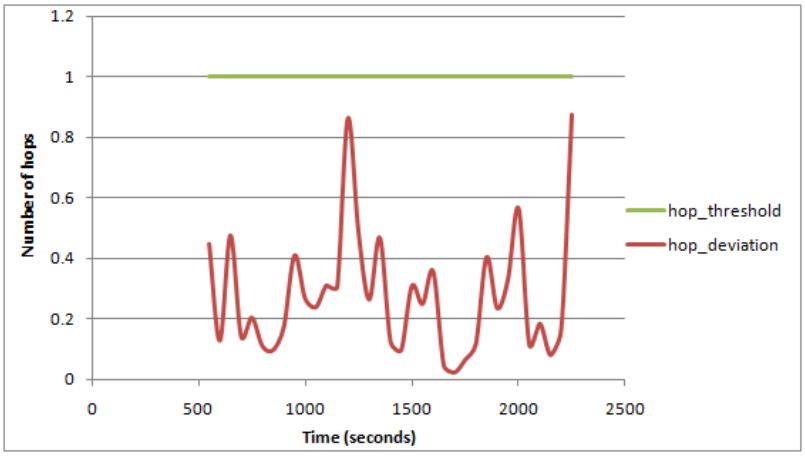

FIGURE 14: HOP COUNT DEVIATION FOR MODE 2

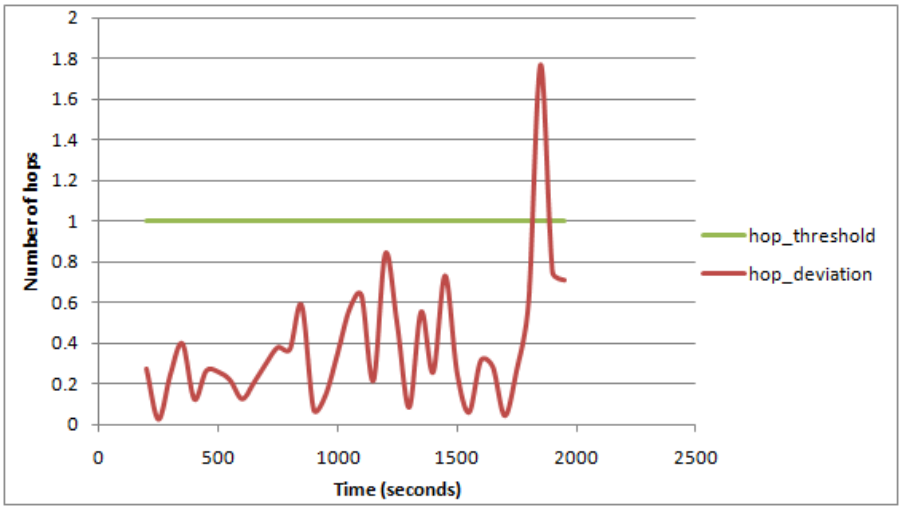

FIGURE 15: HOP COUNT DEVIATION FOR MODE 3

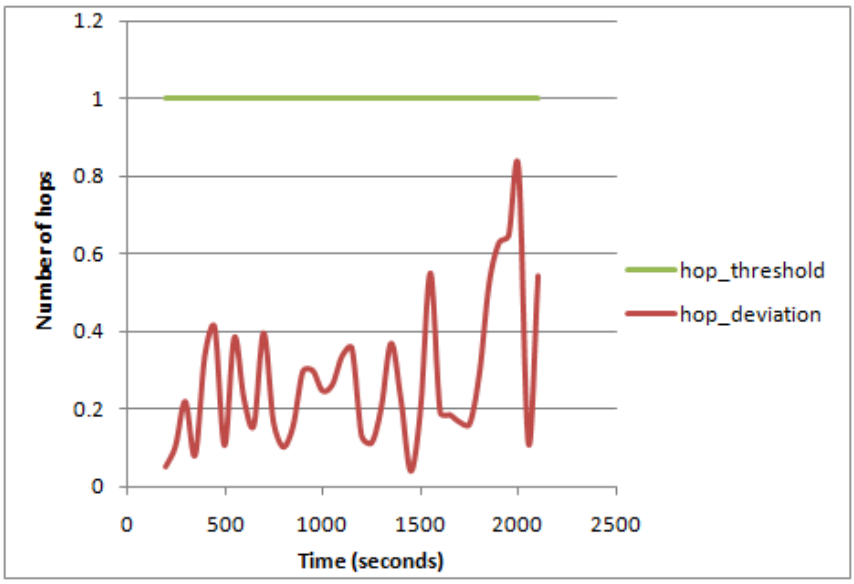

FIGURE 16: HOP COUNT DEVIATION FOR MODE 4

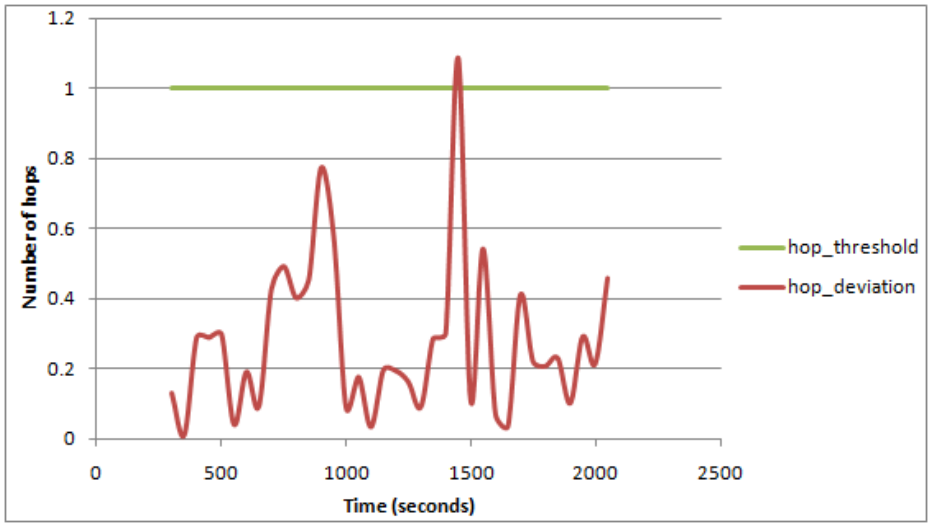

FIGURE 17: HOP COUNT DEVIATION FOR MODE 5

In case of the number of hops, the threshold for hop count deviation is used as 1, since for a sinkhole attack to be effective, the attacker has to advertise a hop count of 1, saying it's the neighbor of base station, or even advertising itself as the base station (hop count = 0), so it can lure all the traffic bound to the base station. Since the input used in the graphs above is typical of an attacker with malicious intent, we see that the deviation remains below the threshold thus indicating that the attacker is still under the deception that the attack is working and hence advertising minimum hop counts from the base station to attract more network traffic.

The success report is generated at each of the deceiving nodes and sent to the DPU for each time period. The DPU uses Dempster-Shafer to combine these reports, along with the uncertainty factors for each of the deceiving nodes for that time period and determines if deception was successful for that particular time period.

The deception success at the DPU calculated uses the reliability of the reporting deceiving nodes while combining the results. The reliabilities of the deceiving nodes were varied to see the following output. When the reliabilities of the deceiving nodes with the range 0-1 were varied, the deception success was found as below for mode 2 and mode 5. The data series below show the deception success curves for when the reliabilities of the deceiving nodes were {100,100,100}, {60, 60, 60} and {30, 30, 30}. The frequency threshold used here is 20% of predicted frequency.

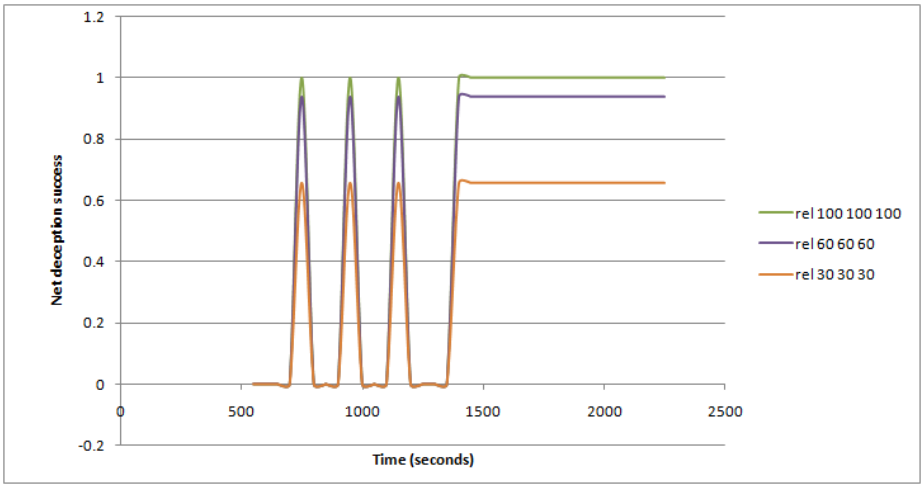

FIGURE 18: DECEPTION SUCCESS BY VARYING NODE RELIABILITIES FOR WORST MODE 2

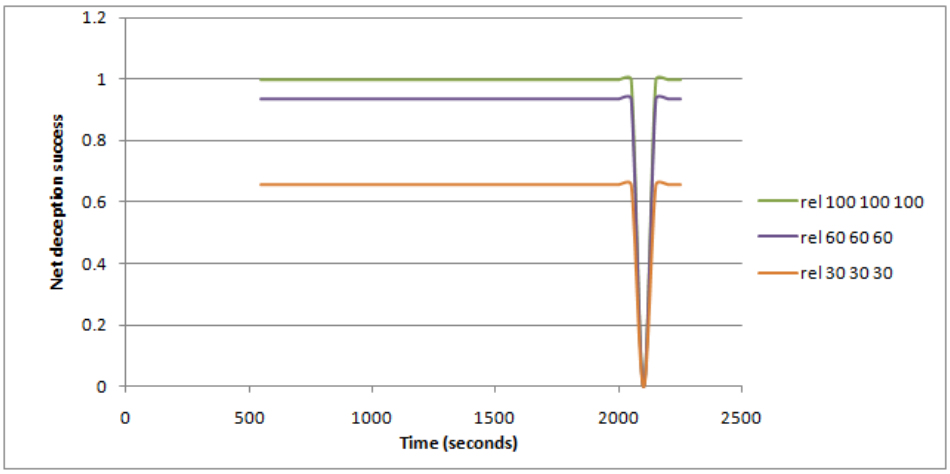

FIGURE 19: DECEPTION SUCCESS BY VARYING NODE RELIABILITIES FOR BEST MODE 2

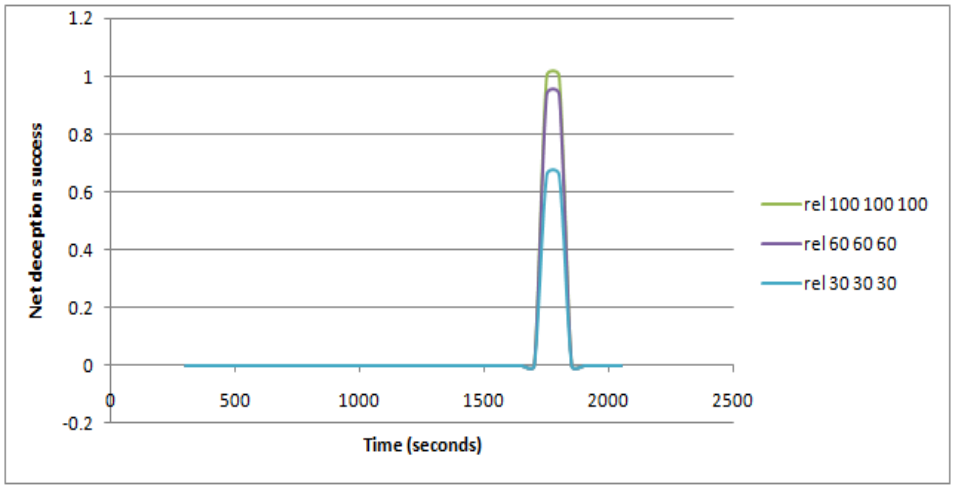

FIGURE 20: DECEPTION SUCCESS BY VARYING NODE RELIABILITIES FOR WORST MODE 5

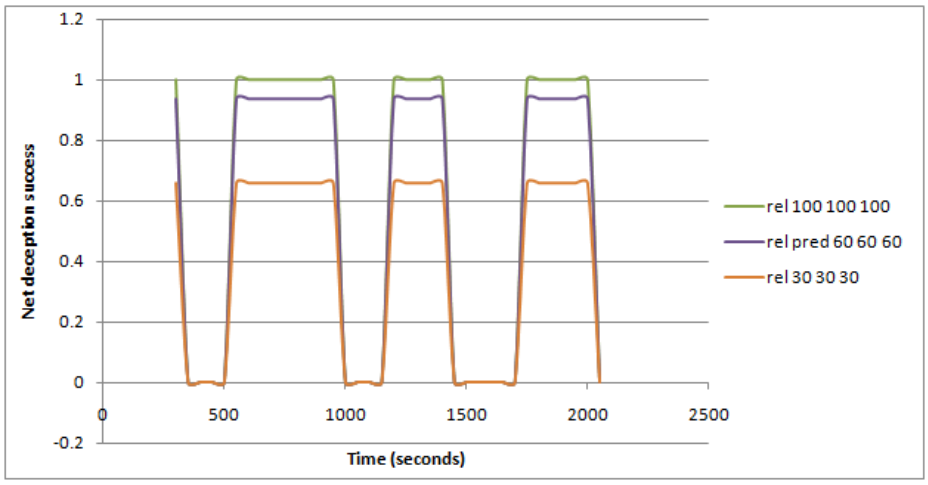

FIGURE 21: DECEPTION SUCCESS BY VARYING NODE RELIABILITIES FOR BEST MODE 5

From the above graphs we see that the overall deception success rate varies when the reliabilities of the deceiving nodes are varied. We see that the deception success curve shuttles between zero (for deception failure) and a constant (determined by the reliabilities assigned to the deceiving nodes). This is because we do not update the reliability of the deceiving nodes with time. The impact of varying reliabilities might be more conspicuous and dynamic with the passing of time if the reliability of each of the deceiving nodes is updated according to the validity of the report given by each of the deceiving nodes so far. Another observation is that the best cases have higher frequency of deception success than the worst cases for the respective modes.

This kind of deception success analysis can be done for many other DoS attacks, wherein the parameters observed would be different as relevant to that particular attack. Each parameter observed needs the attacker behavior and the predictions for that parameter. The predictions should be based on the progressive learning of the attacker's traffic pattern. This will help to catch any sudden changes in the attacker's pattern.

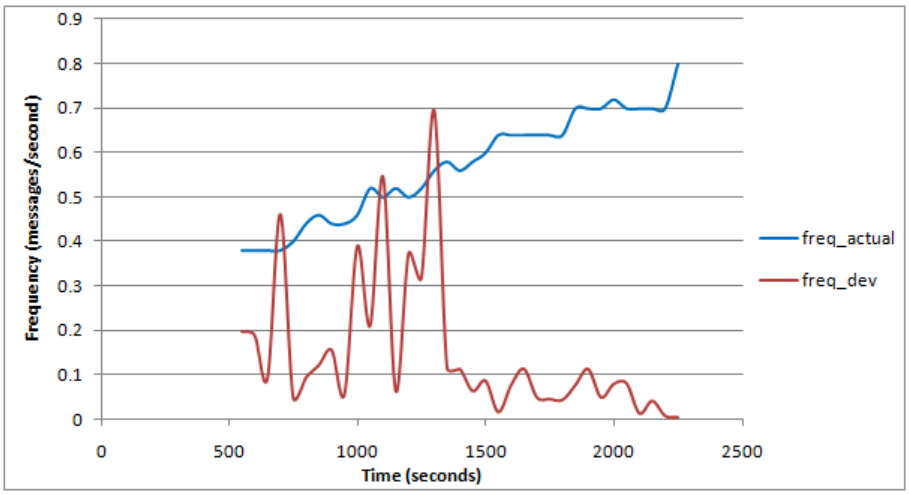

FIGURE 22: ATTACKER PATTERN AND DEVIATION IN WORST CASE MODE 2

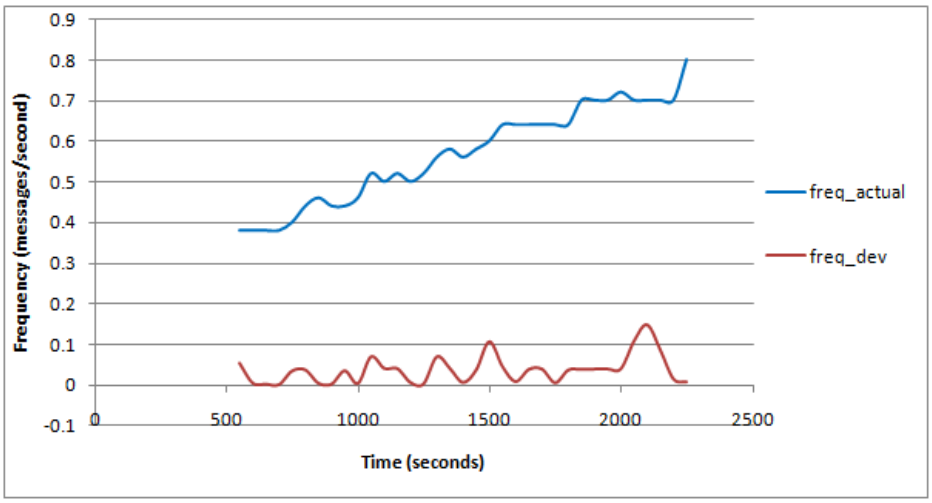

FIGURE 23: ATTACKER PATTERN AND DEVIATION IN BEST CASE MODE 2

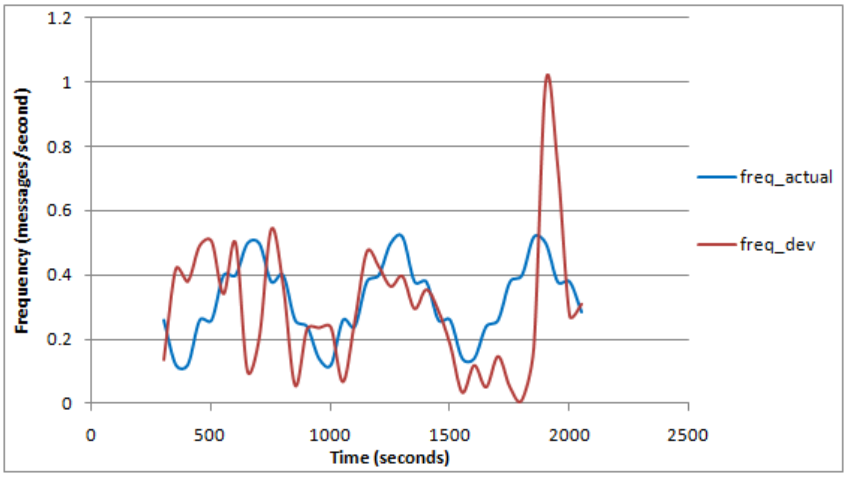

FIGURE 24: ATTACKER PATTERN AND DEVIATION IN WORST CASE MODE 5

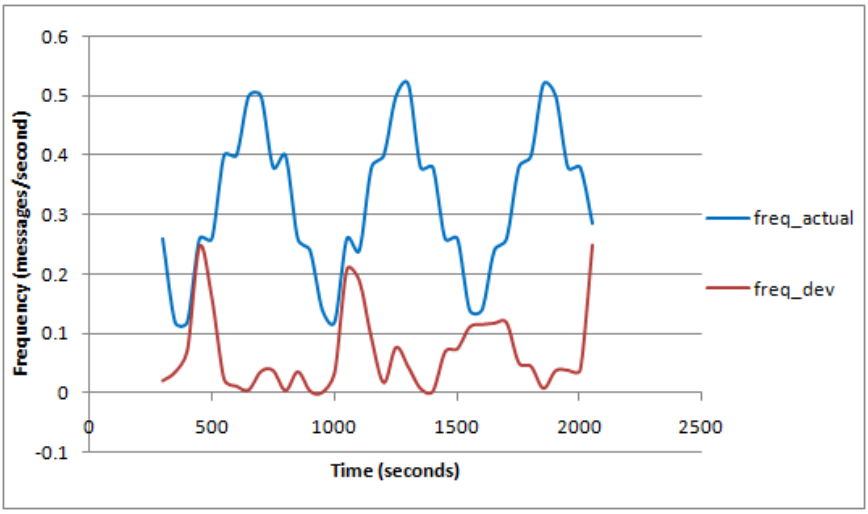

FIGURE 25: ATTACKER PATTERN AND DEVIATION IN BEST CASE MODE 5

In the graphs above, we can see that there is a spike in the deviation curve if the attacker pattern changes in direction or in quantity, i.e., when there is a sudden increase or decrease in the frequency of messages from the attacker or when an upward climbing frequency curve drops suddenly or vice versa. Thus any change in the attacker's behavior that is not typical of an attacker is detected with a fluctuation in deviation. The consideration of uncertainty values at previous time points helps in this regard.

Another point to note is that for the same attacker input for worst and best cases, in mode 2 or 5 above, the deviation varies. This is the effect of prediction on the deviation. The more accurate the prediction is, the less deviation is shown. Thus the best case shows less deviation than the worst case in the above graphs.

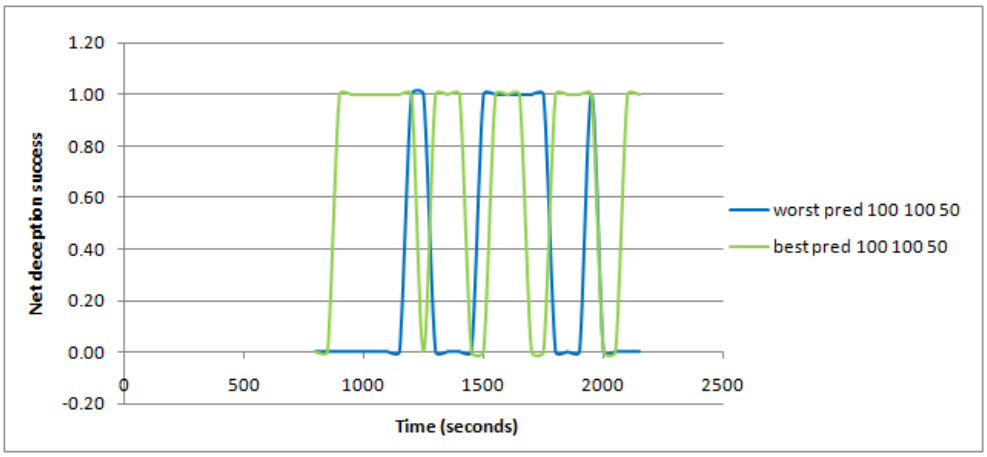

FIGURE 26: DECEPTION SUCCESS WITH BEST AND WORST PREDICTIONS

In the graph above we plot the deception success for the best and worst predictions for mode 1. We see that the deception success rate is higher with the best prediction than the worst prediction. This is self-explanatory, since the closer our predictions are to the attacker's behavior, the higher the success rate. Since the best case predictions are made after frequently referencing the attacker traffic pattern, any sudden changes in the attacker traffic pattern is captured and future predictions are made accordingly. This helps improving the deception success considerably, unlike in the case of worst case predictions.

From the results obtained from the simulation, we believe that our approach proposes a methodology to measure deception success in detail. The uncertainty, which is typical in a wireless sensor network under attack, is also incorporated into measuring deception success.

CHAPTER V

CONCLUSION

In this thesis, an algorithm to analyze and measure of success of deception employed in a wireless sensor network is proposed. The thesis is based on [3], where the deception framework against DoS attacks was developed. We build on top of this by measuring the effectiveness of such a deception framework, using a sinkhole attack. A case study of sinkhole attack is taken and the parameters crucial in recognizing the attack are identified. The prediction based on the attacker's history of behavior is compared against the actual, and the discrepancy is used to evaluate the deception success or failure thereof. The uncertainty of the intruder's intentions are incorporated and updated with time. When the uncertainty increases beyond a point, it causes deception failure. It could mean the intruder moved to another part of the network and hence is not behaving as predicted. We also show the reliability of the deceiving nodes as a part of the end result, since in a WSN with an ongoing attack, that is a major concern.

We learned from the results that when predictions of an attacker's behavior are made, frequently referencing the attacker's progressive behavior and updating the prediction algorithm accordingly helps in increasing the deception success greatly. Moreover the closer our predictions are to the actual attacker traffic, the less the uncertainty is, regarding the attacker intentions. This in turn reduces the net deviation aiding deception success. Any erratic behavior that is not typical of an attacker such as a sudden increase or decrease in the observed parameter of the attacker traffic is captured and reflected in the deviation value, which may be an indication of deception failure. The priority given to the uncertainty in the final deviation value can be varied.

We observed that an improvement would be to make the end result of deception success more dynamic. This can be achieved if the reliabilities of the deceiving nodes are managed and updated regularly with respect to how much the report given by the deceiving node is close to the overall output. Further work can be done to simulate the compromise of a deceiving node by feeding an input not typical of an attacker to that node and observe its effect on the frequency deviation, hop count deviation and the overall output of deception success. Also the work done in this thesis is an ongoing calculation of the deception success at fixed time periods, regardless of whether the deception is a success or a failure. Further studies can be done to estimate a cut-off beyond which

the deception failure is not tolerated and the system reverts to other means of defense to prevent the propagation of attack.

This methodology can be applied to any DoS attack, by identifying attacker's traffic pattern and its parameters crucial to the attack. Although the parameters considered vary with each DoS attack, it lays the general foundation for an effective counter-counter-deception. Although the process of tracking every single parameter critical to the attack and getting predictions for each of them may seem too intensive, it becomes necessary, since the attacker faces the additional challenge of detecting improved deception.

These techniques of counter-counter-deception may induce an "arms race" with attackers continually finding ways to detect improved deception. But for the same reason that anti-virus software is not obsolete because of the viral "arms race", deception improvements provide further impediments and complexity for the attacker, and benefit security in the long run.

REFERENCES

[1] Rowe N. C., "Measuring the Effectiveness of Honeypot Counter-Counterdeception", Proceedings of the 39th Hawaii International Conference on System Sciences, Vol. 6, pp. 129c, 2006.

[2] Shiue L. M. and Kao S. J., "Countermeasure for Detection of Honeypot Deployment", Proceedings of the International Conference on Computer and Communication Engineering, pp. 595-599, 2008.

[3] Zhang R., Thomas J. P. and Mulpuru V. M., "Deception Framework for Sensor Networks", Third International Conference on Security and Privacy in Communications Networks and the Workshops, pp. 361-369, Sep. 2007.

[4] Krontiris I., Giannetsos T. and Dimitriou T., "Launching a Sinkhole Attack in Wireless Sensor Networks; the Intruder Side", IEEE International Conference on Wireless & Mobile Computing, Networking and Communication, pp. 526-531, 2008.

[5] Rowe N. C., "Designing good deceptions in defense of information systems", 20th Annual Computer Security Applications Conference, pp. 418-427, 2004.

[6] Rowe N. C., "A model of deception during cyber-attacks on information systems", IEEE First Symposium on Multi-Agent Security and Survivability, pp. 21-30, 2004.

[7] Braun, J., "Dempster-Shafer Theory and Bayesian Reasoning in Multisensor Data Fusion", Sensor Fusion: Architectures, Algorithms and Applications IV; Proceedings of SPIE 4051, pp. 255–266, Apr. 2000.

[8] Cremer F., den Breejen E., Schutte K., "Sensor Data Fusion for Antipersonnel Land Mine Detection", Proceedings of EuroFusion98, pp. 55–60, Oct. 1998.

[9] Rowe N. C., Custy J. E. and Duong B. T., "Defending Cyberspace with Fake Honeypots", Journal of Computers, Vol. 2, No. 2, pp. 25-36, Apr. 2007.

[10] Rowe N., and Rothstein H., “Two Taxonomies of Deception for Attacks on Information Systems,” Journal of Information Warfare, Vol. 3, No. 2, pp. 27-39, Jul. 2004.

[11] Chen T. M. and Venkataramanan V., “Dempster-Shafer Theory for Intrusion Detection in Ad Hoc Networks”, IEEE Internet Computing, Vol. 9, No. 6, pp. 35- 41, Nov. 2005.

[12] McCarty B., "The Honeynet Arms Race”, IEEE Security and Privacy, Vol. 1, No. 6, pp. 79-82, Nov. 2003.

[13] Krawetz N., "Anti-Honeypot Technology", IEEE Security & Privacy, Vol. 2, No. 1, pp. 76-79, Jan. 2004.

[14] Wood A. D. and Stankovic J. A., “Denial of Service in Sensor Networks”, Computer, Vol. 35, No. 10, pp. 54-62, Oct. 2002.

[15] Cohen F., “A Mathematical Structure of Simple Defensive Network Deceptions”, Computers and Security, Vol. 19, No. 6, pp. 520-528, 2000.

[16] Fowler C. A. & Nesbit R. F., “Tactical Deception in Air-Land Warfare”, Journal of Electronic Defense, Vol. 18, No. 6, pp. 37-44 & 76-79, Jun. 1995.

[17] Koks D. and Challa S., “An Introduction to Bayesian and Dempster-Shafer Data Fusion”, DSTO Systems Sciences Laboratory, Commonwealth of Australia, AR No. AR–012–775, Nov. 2005.

[18] Zhu L., “Sinkhole Attack Simulation in Sensor Networks Using TOSSIM”, Private communications.

[19] Ravindran A., “Deception in Wireless Sensor Networks – Predicting Intruder Behavior”, M. S. Thesis, Department of Computer Science, Oklahoma State University, Jul. 2009.

[20] Spitzner L., Honeypots: Tracking Hackers, Addison-Wesley Professional, 2002

[21] The Honeynet Project, Know Your Enemy, http://www.honeynet.org [last accessed - Mar 15, 2009]

VITA

Vanitha Gopinath

Candidate for the Degree of

Master of Science

Thesis: SUCCESS ANALYSIS OF DECEPTION IN WIRELESS SENSOR NETWORKS

Major Field: Computer Science

Biographical:

Personal Data:
Born in Palakkad, Kerala, India on August 18th, 1977.

Education:
Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in May, 2010.
Received Bachelor of Science in Electrical and Electronics Engineering from Madurai Kamaraj University, Madurai, TamilNadu, India in 2000.

Experience:
Worked as a Senior Systems Executive with eFunds International India Ltd, Chennai, from August 2000 to February 2003. Worked in various development, maintenance and testing projects of Check Order Capturing and Processing systems. Worked briefly as Data Quality Analyst at the Deluxe Headquarters (eFunds' client), Minneapolis, MN, from Apr 2002 – Jul 2002 in the testing module of the ETL project to build operational data warehouse for the Deluxe Corporation.

Name: Vanitha Gopinath Date of Degree: May, 2010

Institution: Oklahoma State University Location: Tulsa, Oklahoma

Title of Study: SUCCESS ANALYSIS OF DECEPTION IN WIRELESS SENSOR NETWORKS

Pages in Study: 39 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study:

Wireless Sensor Networks face mutually conflicting purposes, to provide high security to the network while conserving its limited resources. Although much research has been done in this area in the past, the problem of responding to an attack has not received much attention. Deception is one approach to respond to an attack that enables corrective measure to keep the adversary at bay, without alerting the attacker. We focus on measuring how successful this deception process is using Dempster-Shafer theory for combining evidences and handling uncertainty. We identify the parameters concerned for a DoS attack; incorporate uncertainty of the attacker's intention and reliability of the deceiving nodes themselves and attempt to evaluate the deception success depending on the attacker's behavior with time.

Findings and Conclusions:

We show that it is possible to measure deception success by quantifying the attacker's traffic parameters. It also incorporates the uncertainty pertaining to the true intentions of the attacker and the reliability of the deceiving nodes. We find that if the deceiving node reliabilities are updated with time, depending on their reports, the end result could be more dynamic. Moreover we find that the more the attacker's history is referenced for predicting its behavior, the more successful the deception tends to be.

ADVISER'S APPROVAL: Dr. Johnson Thomas