A FRAMEWORK FOR SIGNAL STRENGTH BASED INTRUSION DETECTION SYSTEM FOR LINK LAYER ATTACKS IN WIRELESS NETWORK

by

Chen Guang Li

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

School of Computer Science

at

CARLETON UNIVERSITY

Ottawa, Ontario
January, 2008

Your file Votre référence
ISBN: 978-0-494-36846-6
Our file Notre référence
ISBN: 978-0-494-36846-6

Table of Contents

List of Tables v

List of Figures vi

Abstract ix

Acknowledgements x

Chapter 1 Introduction 1

1.1 Background 1

1.2 Definition of the Problems 4

1.3 Highlight of the Results 4

1.3.1 Enhanced Accuracy of an RSS-Based Localization Model . . . 4

1.4 Outline of the Thesis 6

Chapter 2 Literature Review 8

2.1 Security in WiFi/802.11i 8

2.1.1 WEP and WPA/WPA2 9

2.2 Attack Models 10

2.2.1 MAC Address Spoofing 10

2.2.2 Rogue AP 11

2.2.3 DoS Attack 11

2.2.4 Jamming 13

2.2.5 Session Hijacking 13

2.3 Related Works 14

Chapter 3 The Signalprint Approach 17

3.1 Signalprints 18

3.1.1 The Motivation for Using Signalprints 18

3.1.2 The Definition of Signalprints 19
3.1.3 Signalprint Generation 19
3.2 Review The Signalprint Matching Method 21
3.2.1 Finding Signalprint Matches 21

Chapter 4 An Enhanced Signalprint Method 24
4.1 Classification Model 24
4.1.1 Training Data 25
4.1.2 Classification Models 26
4.1.3 Model Selection 29
4.1.4 More Variables 31
4.2 Accuracy Enhancement 31
4.3 A Framework of NIDS 33
4.3.1 System Structure 34
4.3.2 Data Model 34
4.3.3 Data Collection and Signalprint Generation 35
4.3.4 Localization Module 36
4.3.5 MAC Spoofing Detector 38
4.3.6 Rogue AP Detector 41
4.3.7 Attack Localization and Disaster Recovery 42

Chapter 5 Evaluation of the Enhanced Signalprint Method 43
5.1 Test-bed Setup 43
5.1.1 Hardware 44
5.1.2 Monitors 45
5.1.3 Send and Receive Packets 46
5.1.4 Geometry 47
5.2 Build a Data Set 48
5.2.1 Data Collection 49
5.3 The RSS Variation 49
5.4 Simulation and Evaluation of Data Mining Algorithms 50

5.4.1 Model Building 50
5.4.2 Data Set for Data Mining Models 50
5.4.3 Results and Comparison 53
5.4.4 A Enhancement Method 55
5.4.5 Enhancement Result 58
5.5 Data Mining vs. Signalprint Matching 60
5.5.1 Attack Simulation 60
5.5.2 Evaluation Method 62
5.5.3 Detector Simulation 63
5.5.4 Confidence Interval 66
5.5.5 The Comparison Results 67
5.5.6 The Enhancement Results 70

Chapter 6 Conclusions and Future Initiatives **73**
6.1 The Accuracy Limitation 73
6.2 Enhancement Result 74
6.3 Advantage of RSS-based NIDS 75
6.4 Detected Attacks 75
6.5 Future Initiatives 76

Appendix A Glossary **77**

Bibliography **80**

List of Tables

Table 3.1 *The RSS value reported by five monitors* 20

Table 4.1 *A part of training data examples* 27

Table 4.2 *A confusion matrix for the LDA model.* 32

Table 5.1 *Part of a data mining data set. The Location column contains integer numbers (location code) that represent the actual x-coordinate, y-coordinate, and z-coordinate of a location. We also define integers* $0, 1, 2, 3$ *to represent the four different antenna directions.* 48

Table 5.2 *The performance of LDA, QDA and classification tree model in scenario one.* 53

Table 5.3 *The performance of LDA, QDA and classification tree model in scenario two.* 54

Table 5.4 *A confusion matrix built on the estimation of the QDA localization model and the actual location of each signalprint* 56

Table 5.5 *The performance of LDA, QDA, and signalprint matching detector. The a=4.8 meters, sample size is 10, and confidence level is 95 percent.* 69

Table 5.6 *The performance of LDA, QDA, and signalprint matching detector. The a=2.4 meters, sample size is 24, and confidence level is 95 percent.* 69

Table 5.7 *Evaluation of the enhancement method of attack simulation* . . 71

List of Figures

Figure 3.1 *Five monitors connected to a signalprint generation server. The network includes two APs and three network nodes. Two APs are working on the same channel.* 20

Figure 3.2 *The matching results based on different matching rules of finding max-matches. A line is drawn between the locations that satisfy the max-match rule.* 22

Figure 4.1 *Plot 500 vectors that consist of three classes shown in three different markers in a two-dimensional space. Since there are five variables in each vector, we cast a five-dimensional space to a random two-dimensional space. In the picked two-dimensional space, there is no way to separate the 500 vectors.* 28

Figure 4.2 *A Classification Tree.* 29

Figure 4.3 *The training vectors are cast on a two-dimensional space* $v_1 \times v4$. *The three classes of data are split by using variable* v_1 *and* v_4. *The splitting using variable* v_3 *does not show because the variable* v_3 *is in the space orthogonal to the space* $v_1 \times v4$. . . . 30

Figure 4.4 *Modules in the NIDS and the working flow.* 35

Figure 4.5 *Data model for training and detecting* 35

Figure 4.6 *The training of a localization model* 37

Figure 4.7 *The working process of the localization module* 39

Figure 4.8 *The list forms a normal pattern.* 40

Figure 5.1 *The setup of the experiment. Five monitors and one server are set up in different rooms. A wireless client moves along the hallway.* 44

Figure 5.2 *Geometry of device location.* 45

Figure 5.3 *The SAR operation.* 47

Figure 5.4 *A plot of RSS values generated by one monitor. The histogram shows the distribution of the RSS values* 51

Figure 5.5 *The result of the enhancement algorithm* 59

Figure 5.6 *The evaluation of the data mining method* 63

Figure 5.7 *The evaluation of the signalprint-matching method* 65

Figure 5.8 *Suppose the attack location has location code 5. The attack detector can easily detect suspicious matches. The evaluator computes the TPR and FPR based on the detection.* 65

Figure 5.9 *ROC analysis on LDA, QDA, and signalprint matching method scenario one. The QDA based detector has the best performance because it yields a point (0.02, 0.96), which represents 96 percent TPR and 2 percent FPR, in the ROC space. It is closest to the perfect point in the upper left corner (0, 1). The best operating point of signalprint matching method is (0.18, 0.83), which represents 83 percent TPR and 18 percent FPR. The matching rule at that point is* $maxMatch(S_1, S_2, 3) > 3 \wedge minMatch(S_1, S_2, 8) \geq 0$. 67

Figure 5.10 *ROC analysis on LDA, QDA, and signalprint matching method scenario two. The performances are degraded as the separation distance of nodes decreases. The QDA based detector yields a point (0.043, 0.87), which represents 87 percent TPR and 4.3 percent FPR, in the ROC space. The best operating point of signalprint matching method is (0.35, 0.83), which represents 83 percent TPR and 35 percent FPR. The matching rule at that point is* $maxMatch(S_1, S_2, 3) > 3 \wedge minMatch(S_1, S_2, 8) \geq 0$. . 68

Figure 5.11 *TPR grows rapidly as the group size increasing. Meanwhile, FPR becomes vary low. When the threshold, a=4.8 meters, the QDA-based detector achieves perfect performance with the group size of eight, TPR=1 and FPR=0. When the threshold, a=2.4 meters, the QDA-based detector achieves best performance with the group size of 16.* 70

Abstract

Although a wireless local area network is claimed to be as secure as a wired network after the deployment of the WiFi protected access (WPA) protocol, because of unprotected medium access control (MAC) management messages, a WiFi network is vulnerable to low-layer attacks, such as MAC address spoofing, session hijacking, rogue access point (AP) and various lower-layer denial-of-service (DoS) attacks. Since it is proved that the received signal strength (RSS) value of a received packet is strongly related to the physical location of a sender, we designed a RSS-based network intrusion detection system (NIDS) framework for MAC layer attack detection. The fact is that most attacks that exploit MAC layer vulnerabilities can be detected by comparing the location of an attacker with the location of victim nodes. The core of the NIDS framework is a RSS-based localization model. The model is based on the quadratic discriminant analysis (QDA) data mining algorithm. The choice of the QDA algorithm is based on the analysis of a simulation of three data mining algorithms, which are linear discriminant analysis (LDA), QDA, and classification tree.

To solve the relative high error of the RSS-based localization model, which is also the problem of RSS-based localization methods in other researches, we designed an enhancement method based on signalprints. For a network where the separation distance of any neighboring node is larger than 2.4 meters, the enhanced localization model can distinguish each node with nearly zero error.

Our RSS-based NIDS focuses on MAC address spoofing attacks. The detection of MAC spoofing attacks is very important since it protects the network from the further identity-based attacks and MAC layer DoS attacks. The localization capability also can be utilized to take effective action after attacks. For the detection of MAC address spoofing, our simulation shows that the NIDS achieves 99.2 percent true positive rate (TPR), and 0.4 percent false positive rate (FPR) when the separation distance of any neighboring node is larger than 2.4 meters.

Acknowledgements

I would like to express my gratitude to Professor Michel Barbeau for his guidance and suggestions.

Chapter 1

Introduction

1.1 Background

The combination of low cost and easy to install hardware, available and unlicensed radio spectrum, along with the proliferation of portable computing devices, such as laptops and PDAs, have made wireless local area networks (WLANs) extremely popular. Unfortunately, a WLAN is vulnerable to attacks. From the serious vulnerabilities in Wire Equivalent Privacy (WEP) protocols to various denial of service (DoS) attacks [5] [16], what is the real concern? The IT industry claims that if the most recent security standard is employed, i.e. WiFi/802.11i, WLANs are as secure as wired networks [8]. However, they have also had to admit that, because of their nature, WLAN radio signals penetrate physical boundaries making infrastructures and clients vulnerable.

The truth is that although the serious vulnerabilities in the WEP protocol can be overcome by deploying the WiFi Protected Access (WPA) protocol, the lack of authentication for management frames still makes WLANs vulnerable to several attacks. The attacks include Medium Access Control (MAC) address spoofing, session hijacking and unauthorized access point (AP). Moreover, several types of DoS attacks and jamming attacks are possible since the MAC address is the only identification of a network client from the point of view of the MAC layer. As an example, the de-association/de-authentication attack [5] can easily knock a client off by sending the de-association and de-authentication frames to the AP with an impersonated MAC address.

It is necessary to employ a network intrusion detection system (NIDS) that functions at the MAC layer and at the physical layer, if high security is required. The NIDS should utilize MAC layer features to build attack detectors to detect the attacks that exploit the security flaws in the data link layer. The problem is to determine

what features we should utilize in a NIDS to obtain a high detection rate and a low false alarm rate.

What is a Good Feature?

For anomaly based intrusion detection systems (IDS), a good feature has to be stable during normal activities and sensitive to the anomalies which are caused by attacks. It also should be simple enough so that the normal pattern can be generated quickly and the detection can be efficient. On the other hand, the feature must not be easily manipulated by attackers. What feature is a good one?

Since we are dealing with the weak identification problem in the MAC layer of WiFi networks, a possible resolution is to build a relation between legitimate network nodes and their physical locations. To do so, two typical features that might be used are the received signal strength (RSS) and time of arrival (TOA). The RSS is easy to measure, unlike the measurement of the TOA which requires multiple antennas and a very fine-grained clock. Actually, a lot of research has been done in using the RSS to develop WLAN location estimating mechanisms [38] [36] [19] [18][20]. We believe that the RSS value of a packet can be use to form a good feature for a NIDS protecting a WiFi network from several MAC layer attacks.

The Signalprint Feature

How to build a NIDS for MAC layer by using RSS? Faria and Cheriton [10] have demonstrated the concept of signalprint in their research about the detection of identity-based attacks in wireless networks. The concept of signalprint is introduced to measure the difference between two physical locations of wireless nodes. A signalprint refers to a vector containing RSS values reported by multiple monitors. As a packet is transmitted over the wireless link, it is sensed by monitors within range. The monitors measure the RSS values as well as other useful information, such as the MAC address, and then report that information to a centralized server. The server generates the signalprint using the RSS reported by all monitors according to the source MAC address of packets. The signalprints are related to the locations of wireless network nodes, since the RSS attenuates with distance.

Is the Signalprint a Good Feature?

At least three questions must be answered. First of all, can a signalprint be easily manipulated? The answer is negative. We believed that a signalprint is hard to manipulate due to the way they are produced. A signalprint is based on the RSS measured from multiple monitors, and the monitors are located in different locations. The locations of monitors, their antenna, and their environment affect the value of a signalprint. Unless an adversary takes control of all monitors, factors affecting the value of signalprint cannot be controlled by an adversary.

Secondly, is a signalprint strictly related to the location of a transmitter? If yes, a NIDS based on signalprints can achieve a high detection rate and low false alarm rate on detecting attacks that can be detected based on the physical location of attacker. The answer is that a signalprint does not accurately reflect the locations of transmitters. Some algorithms need to be developed to enhance the performance of a RSS-based NIDS. In their research, Faria and Cheriton [10] concluded that using six monitors, an accuracy of five meters can be obtained. Eiman et al. [9] presented the fundamental limitations of localization using signal strength in indoor environments. They claim that one can expect a 10 ft. accuracy with an 50 percent error and a 30 ft. accuracy with a 3 percent error. Besides, another concern is that the orientation of antennas and the environment of transmitters can affect the value of signalprints greatly [2] and further degrade the performance.

Thirdly, how to develop a simple detection model based on signalprint so that a NIDS is working efficiently? The training process of a NIDS has to be easy and fast, for at least two reasons. First, a client's location may change because of mobility. Secondly, major changes in the environment may require the NIDS to redo the training process. The changes in the environment could invalidate a localization model. The detector training process includes two phases: the training for the localization model and the training for intrusion detection.

1.2 Definition of the Problems

The lack of authentication for management frames still makes a WiFi/802.11 and a WiFi/802.11i network vulnerable to lower layer attacks. In this thesis, we design a framework for an anomaly based NIDS, in which the RSS value (signalprint) is used as majority feature to classify the normal behavior. The proposed NIDS focuses on detecting the MAC layer attacks. In terms of the network environment, we assume indoor WiFi/802.11 or WiFi/802.11i networks, although we believe that our method can be helpful to other wireless network security systems. We also assume that every node has a fixed location or only moves within several certain locations (e.g. office and conference room), and any two neighboring nodes have a certain separation distance (e.g. at least 1.2 meters).

Since the core of the NIDS is a signalprint-based localization model, low accuracy is another problem. To solve this problem, we first explore the relation between the RSS values and the physical location of a node. Then we propose an enhanced localization model to improve the accuracy and reduce the error rate.

1.3 Highlight of the Results

We make the following contributions: We evaluate and compare three data mining algorithms to find the best one to localize nodes in an indoor environment. Based on the analysis of the evaluation, we propose an enhanced localization model to improve the accuracy and reduce error rate of the RSS-based localization model. We define an attack model for WiFi/802.11 and WiFi/802.11i networks and design a NIDS for it. The NIDS is based on the enhanced localization model. The NIDS not only detects the MAC address spoofing attack and a rogue AP attack, but helps with disaster recovery using the attack localization capability.

1.3.1 Enhanced Accuracy of an RSS-Based Localization Model

We propose an algorithm that uses signalprints to distinguish network nodes. The algorithm is based on data mining techniques. The algorithm can work on both a single signalprint mode and a multiple signalprint mode (enhanced mode). When

working in the enhanced mode, the algorithm can tolerate more uncertain factors that usually degrade the accuracy of a RSS-based localization model, for example, the orientation of antennas of nodes. According to the simulation results in Section 5.4.3, when the separation distance is larger than 2.4 meters and the antenna orientation is an unknown variable, the algorithm can accurately (with a zero percent error rate) differentiate network nodes based on 20 signalprints constructed for each node. Using 30 signalprints per node, when the separation distance of network nodes is 1.2 meters, the localization model can distinguish nodes with one percent error rate. If we assume that the antenna orientations of all network nodes are fixed, then the localization model can distinguish network nodes with a zero error rate using only six signalprints constructed for each node when the separation distance of node is 2.4 meters. When the separation distance of nodes is 4.8 meters, using only one signalprint, the location model can differentiate nodes with an error rate nearly zero.

Detection of MAC Layer Attacks

MAC address spoofing attacks can be detected. For an enterprise WLAN, the network users usually have fixed locations. Whenever two nodes with same MAC address appear to be in a different location, the NIDS classify the network nodes appearing in an abnormal spot as malicious. Or if more than one different MAC address appears to be co-located, the NIDS flags the nodes in that location as malicious. In both cases, the detector can also pinpoint the location of the malicious nodes.

Just like detecting the MAC address spoofing attacks, several DoS attacks, such as the de-association/de-authentication attack and the resource depletion attack can be detected. To launch the de-association/de-authentication attack, the attacker first needs to spoof the MAC address of legitimate nodes, and then send de-association/de-authentication frames to the AP. The attacker who launches a resource depletion attack may send request messages using random MAC addresses. Both scenarios can be detected by the proposed NIDS since the NIDS is able to detect MAC address spoofing attacks.

The proposed NIDS also can detect a rogue AP. We assume that all managed APs in a protected network are running during a training period and the location of each

AP is fixed. After a training period, the NIDS registers all MAC addresses and the locations of authorized APs into a pair list. If an unauthorized AP is located, then the NIDS is able to classify the AP as a rogue AP, since either the MAC address is not in the list or the location determined by signalprint appears abnormal according to the normal pattern of NIDS.

Localization and Disaster Recovery

The location prediction capability makes the NIDS efficient in disaster recovery. After DoS attacks and jamming attacks, the detector is capable of pinpointing the location of attackers accurately. With human assistance, the system can rapidly be recovered.

1.4 Outline of the Thesis

The remaining chapters of the thesis are organized as follow. In Chapter 2, The WiFi security issues are discussed, i.e., WEP in WiFi/802.11, WPA/WPA2 in WiFi/802.11i. An attack model is defined. The related countermeasures are introduced, as well as the related works.

In Chapter 3, we define signalprint and analyze the advantages of building a detector based on signalprint. We review the signalprint matching method and illustrate the mix-match and min-match based on our experimental data set.

In Chapter 4, we present a new algorithm based on data mining techniques that enhance the performance of detectors. We propose a framework for an anomaly based NIDS to detect attacks that exploit MAC layer vulnerabilities, such as MAC address spoofing, an rogue AP and DoS attacks. We solve the data training problems and analyze the robustness of the NIDS. Disaster recovery from DoS and jamming attacks are analyzed.

In Chapter 5, we present the details of model evaluations. A test bed is introduced. An analysis is based on the simulation of localization models and attack detectors. Different approaches are compared, such as a signalprint matching based detector and data mining based detector, as well as localization models based on different data mining algorithms.

In Chapter 6, we conclude and review the highlight of the proposed solutions and we give a plan for future work.

Chapter 2

Literature Review

The WiFi security issues are always a serious concern. A WiFi network provides security service by several security protocols, i.e., WEP in WiFi/802.11 and WPA/WPA2 in WiFi/802.11i. Vulnerability was found in WEP data encryption after WiFi/802.11 was released. In 2004, WiFi/802.11i was released. The new security protocols WPA/WPA2 fundamentally change the security architecture and fix the vulnerability existing in WEP. However, we will show that some weaknesses of the new standard can still be exploited by attackers.

One of the weaknesses in the WiFi/802.11i standard is that the MAC layer management messages are unprotected. Based on this weakness, an attack model is defined. The attacks include MAC address spoofing, rogue AP, various DoS attacks, Jamming attacks and session hijacking. Those attacks can be launched in both the WiFi/802.11 and WiFi/802.11i networks. Countermeasures have been proposed by several researchers and the IT industry. But feasible solutions for MAC layer protection are still needed. We focus on the RSS-based NIDS as a solution. The related works include wireless indoor localization techniques, wireless monitoring and identity-based attack detection. We introduce some related works.

2.1 Security in WiFi/802.11i

A priority for WLANs is protecting the confidentiality of the data while on the air and providing authentication for clients and infrastructures. The WiFi/802.11i standard provides enhanced authentication and encryption mechanisms for WiFi networks. Indeed, WiFi/802.11i cures the serious security problems in WEP, which give the previous standard WiFi/802.11 a reputation of a vulnerable protocol.

2.1.1 WEP and WPA/WPA2

WEP was initially developed to secure 802.11 WLANs. WEP provides device or access point authentication as well as message encryption. WEP encrypts data using the RC4 two-way algorithm. The RC4 cipher applies a pre-shared 40 or 104-bit key plus a 24-bit initialization vector (IV) to the data. The IV is a 24-bit number sent in plaintext before the payload. This means a key stream has 2^{24} (16.77 million) variations regardless of whether it is a 40-bit key or a 104-bit key.

The fact is that 16.77 million keys are not enough to provide strong message encryption considering the relatively small payload of packets and the huge amount of data sent over a network [29] [6] [16] [21]. Over a long period of time, a key has to be reused to encrypt millions of packets. Theoretically, even if all packets carried 1500 bytes data, a key stream will be exhausted after 25 GB data is sent (less than 6 hours at a channel speed is 11Mbps). This is not even sufficient for a home network. A hacker can easily perform a brute force dictionary attack [24] [37].

Instead of waiting for days to collect enough IVs, an attacker can actually use some tools, such as Aircrack [32] and Weplab [31]. Those cracking tools can use packet injection to shorten the WEP cracking time. This weakness is caused by the cyclic redundancy check (CRC) algorithm. Since the RC4 cipher uses a XOR operation to encrypt messages, and CRC is not a security mechanism due to its linearity, there is a one-to-one relationship between unciphered messages and ciphered messages. By using this flaw, a 64-bit WEP key stream in 10,000 packets, or 128-bit WEP key stream in 30,000 packets would be enough to break a WEP key [6].

To fix the security problems that exist in WEP, the WiFi Alliance adopted the WiFi/802.11i [27] standard in June 2004. WiFi/802.11i received the commercial name WiFi Protected Access (WPA). WPA implements most of the WiFi/802.11i standard. Unlike WPA2, WPA is compatible with the first generation WiFi equipment. For the WPA architecture, connection security relies on the Temporal Key Integrity Protocol (TKIP), which is considered to be an upgrade of WEP. TKIP also uses the RC4 cipher and the IV field is increased to 48-bit. Moreover, TKIP provides per-packet key mixing and a message integrity check (MIC). TKIP ensures a unique key for each data packet. The IV is no longer sent in plain text, instead, it is encrypted using a

WPA key. The vulnerabilities in WEP are cured for now.

WPA2 implements the final release of the IEEE802.11i standard. WPA2 fundamentally changes the security architecture. The new architecture is called Robust Security Network (RSN). RSN only accepts RSN-capable devices. For the sake of compatibility, IEEE802.11i also defines an architecture called Transitional Security Network (TSN), that accepts both RSN and WEP.

It seems that the only practical vulnerability that can be exploited to break WPA is the WPA/WPA2's Pre-share Key (PSK) [16]. PSK is an alternative to Pairwise Master Key (PMK) generation, which uses an authentication server. PSK is a 256-bit string. PSK could be generated using a passphrase of 8 to 63 characters. Although PSK is not exposed on the network, the derivation of a Pairwise Transient Key (PTK) uses information transmitted in plain text during 4-way handshake. Therefore, the strength of PTK relies only on the PSK, which is generated by a passphrase. This means an attacker can perform a brute force decryption attack using the packets transmitted during a 4-way handshake. However, to successfully crack a PSK, an attacker requires a large amount of 4-way handshake messages, and also requires a weak passphrase (less than 20 characters). One way to speed up the attack is to combine the de-authentication attack to force wireless clients and the AP to repeatedly perform a 4-way handshake [21].

On the other hand, MAC layer management messages in the WiFi/802.11i standard are unprotected. This makes WPA/WPA2 unable to protect the WiFi networks from low-level attacks, such as MAC address spoofing, session hijacking, rogue AP and various lower-layer DoS attacks. [8] [24] [16][37].

2.2 Attack Models

2.2.1 MAC Address Spoofing

The MAC addresses of legitimate nodes can easily be discovered by sniffing the traffic because of unencrypted management frame. It is also not hard for an attacker to change its own MAC address. This means that the MAC address spoofing attack is easy to launch. To launch a MAC address spoofing attack, an attacker has three

motivations: bypass access control lists, impersonate an authenticated user or hide its presence on a network.

We are concerned about MAC address spoofing because MAC address spoofing detection makes it possible to detect other attacks. MAC address spoofing is the first step of all identity-based attacks. In the previous section, we discussed how an attacker can perform packet injection or de-authentication attacks in order to crack WEP or WPA/WPA2 faster. Moreover, various DoS attacks start from MAC address spoofing, such as the de-association/de-authentication DoS [5] and resource depletion attacks [10].

2.2.2 Rogue AP

Another kind of attack we focus on is rogue AP. Rogue AP may be set up by a malicious person or a legitimate user. An attacker could set up a fake AP, which appears as a valid authenticator, to deceive wireless clients in order to derive critical information. On the other hand, legitimate users could set up APs for their own convenience, sharing data, or extending network accesses. Those APs break the security policy that is supposed to control the access of network, and are usually not protected by any security mechanism. The users who set up those APs actually compromise the network security without being aware of doing so [8][7].

2.2.3 DoS Attack

DoS attacks are always an administrator's headache in the computer network world. For the WiFi networks, some new types of DoS attacks can be launched by exploiting the vulnerabilities in the MAC layer, such as de-authentication and de-association attack, power saving attack, distributed coordination function (DCF) attack and resource depletion attack[5] [8] [16][7]. We have observed that an attacker usually has to spoof the MAC address of a legitimate node before the attacker launches these DoS attacks. We use two strategies to mitigate DoS attacks. At first, we rely on the detection of the MAC address spoofing attack to protect a network from various DoS attacks. Second, after a DoS attack happen, we utilize the localization model to help disaster recovery.

De-authentication and De-association Attack

De-authentication and de-association DoS attacks could be easily launched because of unprotected authentication and association management frames. In an infrastructure WLAN, a client must associate with an AP before it can exchange data with other nodes. Before a client is associated with an AP, it must be authenticated first. Several management frames are involved to complete the association between wireless clients and the AP. However, in a WiFi network, management frames are unprotected. An adversary could launch a DoS attack by disturbing the association or authentication between clients and an AP.

Attack on Power Saving

Based on the same strategy, another DoS attack exploits power saving mechanisms. To conserve energy, wireless clients are able to enter into a sleep state. While a client is in the sleep state, the AP buffers all the inbound traffic for the client. Periodically, the client awakens and checks the beacon frames sent by the AP to find out whether it has inbound data. If there is inbound data, the client sends a power-save poll frame to the AP, and then the AP delivers the data for the client and deletes the buffer. An attacker can first perform MAC address spoofing and then send a power-save poll frame to the AP. Consequently, the inbound data destined for the client is lost because the client cannot receive any data while in the sleep state. Moreover, whenever the victim client sends a power-save poll frame to AP, it receives a false negative report from the AP.

Attack on DCF

The DCF can be exploited to design DoS attacks. DCF is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. A wireless node can sense the carrier through the physical layer and network allocation vector (NAV). On each node, the NAV stores the duration that was sent in the request to send (RTS) / clear to send (CTS) frames. The NAV indicates that the medium is idle if its countdown counter is zero. Therefore, a node can only transmit when the counter is zero. By sending malicious RTS frames, an attacker can assert a large duration field

in NAV to monopolize the channel. On the other hand, an attacker may break the rules of CSMA/CA. An attacker can misuse the channels by sending a signal before the end of every short inter-frame space (SIFS) period or by choosing a smaller size of contention window.

Resource Depletion Attacks

Because a MAC address is used as an identifier of a legitimate user, network systems will assign specific network resources to a network user after that user is associated to the AP. An attacker may use random MAC addresses to create associations with an AP in order to flood it, or send request messages to exhaust network resources, such as dynamic host configuration protocol (DHCP) requests [10].

2.2.4 Jamming

Jamming attacks can be achieved not only by using strong noise (physical layer attack), but also by overriding the MAC protocol. For example an adversary can just disregard the MAC protocol and continually transmit on a channel. Actually, there are various attack models that attackers can deploy, for example, attack on DCF, as discussed earlier.

Since jamming is either malicious or unintentional (network congestion), it is not easy to discriminate between congestion or a jamming attack. Our method may not detect a jamming attack but the proposed localization model can help to locate an attacker after the jamming attack is detected (can be detected by using radio spectrum monitoring equipment). [4] [34] [28] [26]

2.2.5 Session Hijacking

By exploiting unprotected management frames, an attacker is able to launch session hijacking attacks. After a legitimate network node has finished the authentication and association process, an attacker may perform possible attacks to knock down the legitimate device, or the attacker may wait until device is in sleep mode. Then, it masquerades as that device to gain possible access to the network. Although an

attacker is not able to read encrypted messages without breaking the data confidentiality and integrity protocols, i.e. WPA, the attacker is able to get access to the network resources. Meanwhile, as we presented earlier, after the successful launch of a session hijacking attack, an attacker can exploit the weakness of WEP to break the WEP key quickly.

To launch a session hijacking attack, an attacker has to masquerade as a legitimate user by spoofing the MAC address, so we rely on the detection of MAC address spoofing to prevent such attacks.

2.3 Related Works

Faria and Cheriton [10] proposed a method to detect identity-based attacks in WiFi/802.11 WLANs by using the RSS. According to their experimental results, the RSS is correlated with the physical locations of network nodes. A concept of signalprint was introduced. A signalprint is a vector that contains the RSS values measured from multiple monitors. The RSS values are reported in decibel milliwatt (dBm), i.e., A RSS of s watts is $10 \times log_{10}(s \times 1000)$ dBm. They searched for max-matches and min-matches of signalprint pairs. First, they defined a threshold value ϵ dBm, then compute the value of vector $M = abs(S_1 - S_2)$, where S_n refers to the signalprint vector of the n^{th} packet. The signalprint vector S is defined as $S = (v_1, v_2, \ldots, v_i)$, where i is the number of monitors. v_i is the RSS value measured by the i^{th} monitor. The max-match is defined as the number of elements in vector M with a value less than ϵ dBm. The min-match is defined as the number of elements in vector M with a value larger than ϵ dBm. For example, $maxMatches(S_1, S_2, 10) = 3$ means that three 10 dBm max-matches are found when comparing signalprint S_1 with S_2. Intuitively, max-matches measure the similarity between two signalprints and min-matches measure the difference of two signalprints. If the number of max-matches is bigger, it is more likely that the two packets related to the signalprints are sent from the same network node. On the contrary, if the number of min-matches is bigger, it is more likely that the two packets related to the signalprints are sent from the different network node.

To minimize false positives of their IDS, they combined max-matches and min-matches to form matching rules. They said that a pair of signalprints *match* if the signalprints satisfy a specified matching rule. For example, a matching rule can be

$$maxMatches(S_1, S_2, 6) > 5 \wedge minMatches(S_1, S_2, 7) = 0$$

This further reduces the number of error matches. With optimized matching rules, their detector can be used to detect resource depletion attacks and masquerading attacks. They drew a conclusion about the relation between signalprints and physical locations. Using six monitors, an accuracy of five meters can be obtained. Their method does not directly rely on the localization of network nodes and they did not indicate whether or not their experimental results considered the change of antenna orientations.

Wireless user localization based on RSS has been proposed by several researchers [14] [36] [30]. Bahl et al. [2] proposed a RSS-based user location and tracking system. In their experiment, they discovered that RSS at a given location varies significantly depending on the antenna orientation. They considered the antenna orientation as a factor in their experiment, which is not clearly mentioned or considered in other experiments. For an acceptable error rate, their localization model is able to estimate a node's location with accuracy of 2.94 meters inside a building.

Elnaharwy et al. [9] made a comparison between several localization approaches based on RSS in WiFi/802.11 networks. They obtained a fundamental limitation for the RSS approach to localization in indoor environments. With an acceptable error rate, the accuracy is about 30ft.

RF fingerprinting (RFF) has been developed to distinguish wireless devices [11][15]. A fingerprint for a wireless device refers to several features, such as phase and amplitude, which are extracted during the period when a device powers up before a transmission. Research has been done into using RFF to detect rogue access points [15]. The advantage of using RFF is that it reflects the unique hardware characteristics of a wireless device and thus cannot be easily forged. However, generating RFFs requires specialized hardware.

Guo et al. [13] proposed an algorithm to detect MAC address spoofing by leveraging the sequence number in MAC frames. The sequence numbers in the MAC frames

sent from the same node are successive. The sequence number ranges between 0 and 4095, whenever a new frame is sent the sequence number is increased by one. In their paper, they analyzed the gaps of sequence numbers and the distance of out of order frames in a sequence of MAC frames sent from a wireless node. They drew the conclusion that if a gap is larger than 3 frames, the distance between two out of order frames will never be bigger than 4.

Several researchers addressed problems in wireless monitoring [35] [17] [3]. The problems included the location of monitors, the poor performance of wireless monitoring, and monitoring multiple channels. A good idea from [1] is using a dense array of inexpensive radios (DAIR). DAIR is based on the observation that there are plenty of desktop computers with good wired connectivity and spare CPU resources in most enterprise environment. DAIR makes use of existing desktop computers with USB-based wireless adapters as monitors to build a low cost wireless monitoring system and also provide a better performance.

Chapter 3

The Signalprint Approach

The detection of a MAC address spoofing attack is a very important task that a MAC layer NIDS must perform. If a NIDS is able to detect a MAC address spoofing attack, not only all identity-based attacks can be detected, but various DoS attacks can be prevented. Without the possibility of MAC address spoofing attacks, an attacker is unable to launch a de-authentication, de-association or packet injection attack in order to break a WEP key in a short time, and it is believed to be difficult to break a WPA key [21].

In order to detect a MAC address spoofing attack, we need a method for distinguishing wireless devices or a way to determine which wireless device send out a specific packet. The RFF is able to reflect the unique hardware characteristics of a wireless device but specialized hardware is required to create the RFF. The consistency of sequence numbers in transmitted frames can be used to differentiate the frames sent by different wireless devices. A big gap in the sequence numbers of frames that are supposed to be sent from a network node may indicate a frame with a spoofed MAC address. However, a normal frame loss creates a high false alarm rate with a detector based on frame sequence number. Some other difficulties with this approach are the device reboot and forged frame sequence numbers.

The RSS of a packet is related to the location of the sender. The location information can be used to detect MAC address spoofing. Whenever two nodes with identical MAC address appear to be at different locations, or several frames with different MAC addresses appear to be sent from the same source, a MAC address spoofing attack may be suspected. On the other hand, the location information can be used to detect a rogue AP, since the location of authorized APs is known and fixed. Besides, location information reveals the positions of attackers, or the nodes that cause problems.

However, the RSS does not accurately reflect the locations of wireless devices. The inaccuracy may degrade the performance of a RSS-based NIDS. A high false alarm rate makes a NIDS unfeasible, and also can be utilized by an attacker to launch a squealing false positive attack [23]. To solve this problem, a suitable RSS-based localization algorithm is needed.

3.1 Signalprints

Although there are several methods available for indoor wireless device localization, our approach is based on signalprints that are generated by the RSS value collected from multiple monitors. The definition of signalprints is given in this section as well as the way a signalprint is generated from RSS values.

3.1.1 The Motivation for Using Signalprints

Research has been done on localization based on RSS. First of all, the experiments have proved that the RSS is strongly related to the locations of wireless devices [36] [10]. Secondly, most localization algorithms are based on the RSS reported from multiple monitors. It is also possible to localize a wireless node using multiple observations from that wireless node. Instead of using the RSS of packets sent from the node, Ladd et al. [20] proposed a localization model that uses the RSS of packets sent by several different APs. Also, a localization model based on the RSS that is reported from a single WiFi AP was proposed by Zaruba [38].

Our proposal is based on signalprints generated by the RSS value reported from multiple monitors. There are two reasons. First, we aim to design a NIDS based on RSS for enterprise WLANs. Data collection from several monitors is much easier than data collection from each network device. Regarding a localization technique based on a single AP, a single AP may not cover the whole area of a WLAN. Secondly, the RSS reading from multiple monitors is hard to manipulate. The monitors are located in different locations and assumed to be secured. The reading of the RSS for each packet is unpredictable because a lot of variables affect the value of the RSS, such as the walls, the antenna orientation, and so on. Unless an attacker takes control of all monitors, the RSS readings from multiple monitors can not be forged

and therefore a signalprint based on the RSS value from multiple monitors satisfies one of the requirements to be a good feature of a NIDS.

3.1.2 The Definition of Signalprints

A signalprint refers to a vector $S = (v_1, v_2, \ldots, v_n)$. The n is the number of monitors. We define v_i as the RSS value reported by the i^{th} monitor. A RSS value is reported in decibel milliwatt (dBm). We assign -100 dBm as a default RSS value when a monitor does not capture a packet observed. Most of the wireless network adapters can not pick up a signal when the RSS is below -90 dBm. We use S_j refers to the signalprint of a packet j.

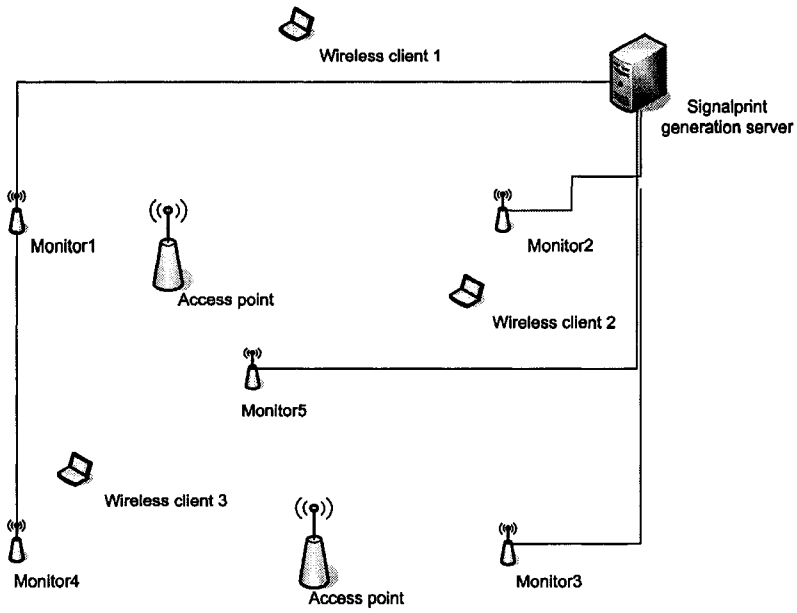
3.1.3 Signalprint Generation

After a packet is captured by a monitor, the RSS value, the source MAC address, destination MAC address, and frame sequence number of the packet are sent to a signalprint generation server. The server generates a new signalprint when it receives several RSS values from monitors for the same new packet. To determine whether a RSS value belongs to a specific signalprint S_j, we define a time out window T. Within T, a RSS value with same source MAC address and frame sequence number, which are reported along with the RSS value, belongs to the same signalprint. A security server puts the RSS value reported by the monitor i in a corresponding position to the signalprint S_j. We assume that in the time window T, the sequence number in the frames sent out by a sender will not repeat. Usually, whenever a new frame is sent the sequence number is increased by one, and the sequence number ranges between 0 and 4095.

Meanwhile, a signalprint is tagged as matured after it has been created T times. A given packet cannot be captured by a monitor twice. Every time a monitor, i, reports the RSS value for a packet, if the security server cannot find any immatured signalprint that has the same source MAC address and frame sequence number, then a new immatured signalprint for that packet is created.

As an example, suppose we deploy five monitors for signalprint generation. The

Figure 3.1: *Five monitors connected to a signalprint generation server. The network includes two APs and three network nodes. Two APs are working on the same channel.*



network structure is shown in Figure 3.1. Three packets are sent out by three wireless devices with different locations. The RSS values are shown in Table 3.1. Because the packet sent out from the wireless client1 may not be captured by monitor3 and monitor4 since they are out of range, the elements of v_3, v_4 in S_3 are equal to the default value -100 dBm. The generated signalprints are: $S_1(-67, -59, -73, -70, -73)$, $S_2(-53, -52, -72, -80, -69)$, $S_3(-56, -60, -100, -100, -72)$.

Table 3.1: *The RSS value reported by five monitors*

Packet	RSS 1	RSS 2	RSS 3	RSS 4	RSS 5
1	-67	-59	-73	-70	-73
2	-53	-52	-72	-80	-69
3	-56	-60	-	-	-72

3.2 Review The Signalprint Matching Method

The concept of signalprint max-matches and min-matches are introduced by Faria and Cheriton [10]. The min-matches measure the difference between two signalprints and, therefore, can be used to detect packets from different nodes. The max-matches measure the similarity of two signalprints and can be used to detect packets from a single node. Moreover, using min-matches and max-matches together as matching rules can make the detection even more precise.

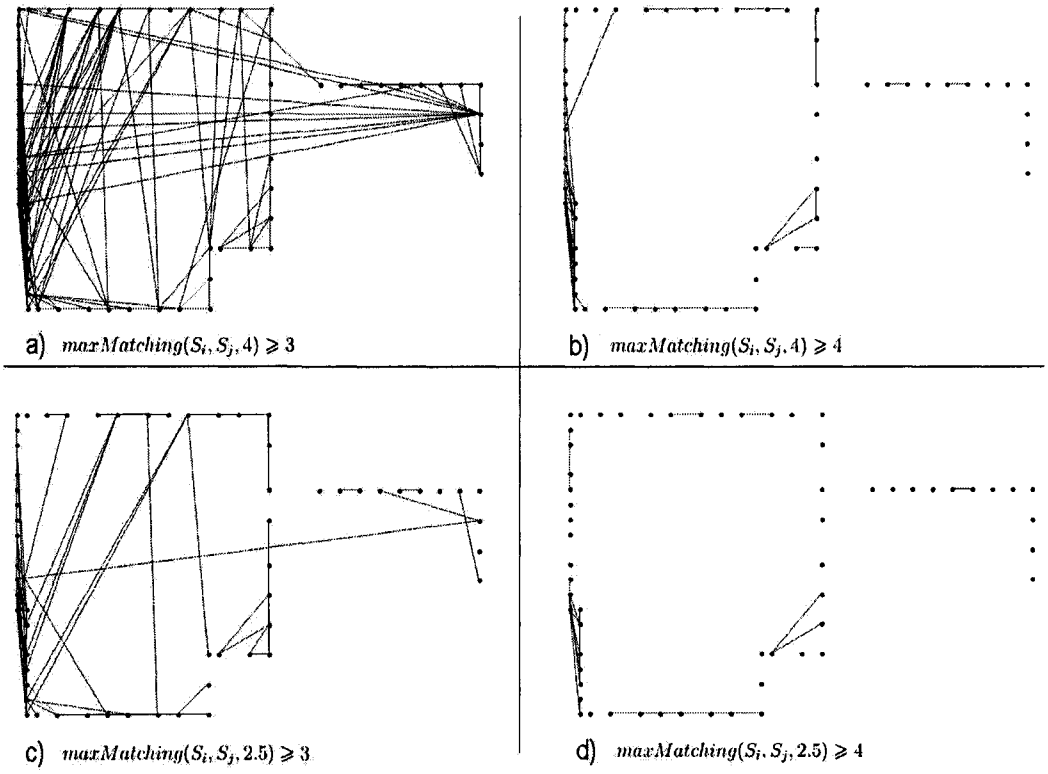
3.2.1 Finding Signalprint Matches

The signalprint vector S is defined as $S = (v_1, v_2, \ldots, v_i)$, where i is the number of monitors. To find max-matches, first define a threshold value ϵ, then compute the value of vector $M = abs(S_1 - S_2)$, where S_n refers to the signalprint vector of the n^{th} packet. The max-match is the number of elements in vector M with a value of less than ϵ. For example, we have two signalprints generated by five monitors, $S_1 = (-72, -60, -80, -74, -80)$ and $S_2 = (-70, -90, -78, -80, -60)$. Then we have $M = (2, 30, 2, 6, 20)$. If we define the threshold value $\epsilon = 10$, then there are three max-matches between S_1 and S_2. This means there are three elements in signalprint S_1 and S_2 that the difference between them are less than 10 dBm. So, the measurement of max-matches is useful when finding signalprints related to the packets sent from the same sender. To find min-matches, first define a threshold value ϵ, then compute the value of vector $M = abs(S_1 - S_2)$. The min-match is the number of elements in vector M which value is at least ϵ. For example, with the same two signalprints and the same threshold value $\epsilon = 10$, then there are two min-matches between S_1 and S_2. This means there are two elements in signalprint S_1 and S_2 and that the difference between them is more than 10 dBm. So, the measurement of min-matches is useful when finding signalprints related to the packets sent from distinct nodes.

Figure 3.2 shows the matches, which satisfy a specific max-matches rule, found between the signalprints related to 59 locations. The signalprints were the part of our simulation data set that was generated to evaluate our proposed RSS-based NIDS (the details are shown in section 5.2). The matches of the signalprints show the relation between a node's physical location and the signalprints generated for the packets sent

from it. The strong relation makes it possible to distinguish the nodes according to the signalprints of packets they send out. Meanwhile, Figure 3.2 shows how the different matching rules affect the matching results.

Figure 3.2: *The matching results based on different matching rules of finding max-matches. A line is drawn between the locations that satisfy the max-match rule.*



In Figure 3.2, a line is drawn between two locations if there is a match. This indicates that we have found the required number of max-matches between the signalprints produced for these locations. Instead of signalprints for individual packets, we generate 59 new signalprints for each location by using the median RSS value of all signalprints related to each location. Since our purpose is to find an optimal matching rule to distinguish individual nodes in different locations, the matching rule requires multiple max-matches with very low ϵ threshold. A better matching rule

should avoid the *long distance matches*. The *long distance matches* are matches between the nodes that have large separation distance, and indicate the nodes cannot be distinguished according to the matching rule. This is because that the max-matching rule measures the similarity of signalprints. If there are multiple max-matches between two signalprints, those signalprints are similar according to the max-matching rule. In Figure 3.2, a) shows the matches that are found when the $\epsilon = 4$ dBm and the number of max-matches is three. Under this matching rule, we can see a lot of *long distance matches*. The *long distance matches* can be avoided through the adjustment of matching rules. First, we increase the number of max-matches required. In Figure 3.2, b) shows the result after changing the max-match number to 4. The number of *long distance matches* is decreased.

Another way to reduce the number of *long distance matches* is to decrease the threshold ϵ. In Figure 3.2, c) uses the matching rule that $\epsilon = 2.5$ dBm and the required number of max-matches is three. In Figure 3.2, d) shows the result when we decrease ϵ and increase the required number of max-matches at same time. $\epsilon = 2.5$ dBm and the number of max-matches is 4. There are almost no *long distance matches* after the adjustment.

The signalprint matching method proves that the signalprints are strongly related to wireless devices physical locations and it is possible to identify a wireless device based on the signalprints of the packets sent by that device. However, according to the simulation in Section 5.5 and the research by Faria and Cheriton [10], the accuracy of a localization model based on signalprint matching method is not good enough to build a NIDS on it.

In the next section, we propose localization models based on different data mining algorithms and present a solution to enhance the accuracy of the localization model. The motivation is to make a RSS-based NIDS more sensitive to possible attacks and decrease the false alarm rate.

Chapter 4

An Enhanced Signalprint Method

According to the research that has been done using the RSS to localize a wireless node, the signalprint of a packet is strongly related to the location of the sender. Using this relation, we use data mining techniques to build a classification model to estimate the location of a wireless device according to the signalprints of the packets it sends.

4.1 Classification Model

Classification is a kind of supervised learning. Classification creates a function (classification model) from training data. The function determines the *class* of the input vector. A training data set consists of input vectors and *class labels* for each vector. A classification process includes five steps:

- determine the type of training data,
- generate training data set,
- determine the input vectors,
- determine the data mining algorithm, and
- optimize and validate the classification model.

Before doing classification, we determine what kind of training data is collected. A proper training data set consists of desired *class labels* and potential features that help build an accurate classification model. In our case, the training data set consists of RSS values, locations of wireless devices, and antenna orientation information.

To generate a training data set, we not only collect the RSS values of a packet reported by multiple monitors, but utilize packets containing more information to satisfy the requirements of training data generation. The information contains location

information, antenna orientation and serial numbers, and so on. We develop a tool called SAR to generate the network traffic we need and extract the information from captured packets. During the training mode, the SAR running on network nodes send packets to an AP. The monitors will capture the packets, extract the useful information, and send it to a training data set generator of the classification model.

During the testing mode, since we are concerned about the relation between the RSS values of a packet and the location of the sender, the vectors input to the classification model are signalprints. For accuracy and efficiency of a classification model, an input vector should contain features that represent an input object well and the number of features should not be too big. On the other hand, adding more information to input vector can improve the accuracy. But in practice, the information we can get usually is limited.

Determining a suitable data mining algorithm for a RSS-based NIDS is a trade-off between accuracy, efficiency, and time of model generation. Some other factors may also be involved, such as the computational complexity and size of training data sets. The classification algorithms we use include linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and classification tree.

4.1.1 Training Data

We first define a training data set for the data mining model. A training vector $T = (S, i, o)$, where $S = (v_1, v_2, \ldots, v_n)$ is a signalprint vector, v_k is the RSS value for a packet reported by the k^{th} monitor. The n is the number of monitors. Variable i is in a finite space $I = (i_1, i_2, \ldots, i_m)$ and is used as a *class label* when training the classification model. The i could be the MAC address of nodes or the location of nodes. The m is the number of nodes in a WLAN. The o is a variable that represents the antenna orientation of each wireless node. In future work, other variables may be added, such as time. So we can define a training data set consisting of vectors $T = (v_1, v_2, \ldots, v_n, i, O)$. O is an optional vector $O = (o_1, o_2, \ldots, o_j)$ for variables that may affect the value of signalprints.

4.1.2 Classification Models

Linear and Quadratic Discriminant Analysis

Fisher has originally defined discriminant analysis [12]. LDA is used for classification when the input variables are in two or several classes. To discriminate two classes of data, Fisher's idea is to transform multivariate observations X into univariate observations $Y = w^T \cdot X$ by using linear combinations of X, such that in observations Y, populations in different classes, c_i, $i = 1, 2$, are separated as much as possible. The approach is finding a w to maximize the ratio of *between class variance* to *within class variance.*

Suppose two classes of observations have means:

$$\mu_i = E(X|c_i), \text{ for i=1,2.}$$

and the covariance matrices;

$$\Sigma_i = E(X - \mu_i)(X - \mu_i)^T, i = 1, 2$$

then $Y = w^T X$ and

$$S = \frac{\text{the between class variance of Y}}{\text{the within class variance of Y}} = \frac{(w \cdot \mu_{2Y} - w \cdot \mu_{1Y})^2}{w^T \Sigma_{2Y} w + w^T \Sigma_{1Y} w} = \frac{(w \cdot (\mu_{2Y} - \mu_{1Y}))^2}{w^T (\Sigma_{1Y} + \Sigma_{2Y}) w}$$

We assume the two class populations have a common covariance matrix, i.e. $\Sigma = \Sigma_1 = \Sigma_2$. We have

$$S = \frac{(w \cdot (\mu_{2Y} - \mu_{1Y}))^2}{w^T \Sigma w}$$

It can be shown that S is maximized by $w = \Sigma^{-1}(\mu_2 - \mu_1)$. After the transformation, the average of the two univariate means for Y, m can be used as a separator of two classes.

$$m = 0.5(\mu_{1Y} + \mu_{2Y}) = (\mu_1 - \mu_2)^T \Sigma^{-1} \left[\frac{(\mu_1 + \mu_2)}{2}\right]$$

We can classify an observation x_0 to class c_1 if $(\mu_1 - \mu_2)^T \Sigma^{-1} x_0 \leq m$ and to class c_2 otherwise .

If we do not hold the assumption that two class populations have a common covariance matrix, we have a quadratic discriminant function. x_0 classifies as c_1 if

Table 4.1: *A part of training data examples*

v_1	v_2	v_3	v_4	v_5	i	o
-52	-58	-70	-75	-62	22Loc	0
-52	-58	-71	-75	-62	22Loc	0
-53	-51	-72	-81	-69	15Loc	0
-53	-53	-74	-81	-69	15Loc	0
-65	-53	-77	-72	-73	05Loc	0

$$(\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})x_0 \leq 0.5x_0^T(\Sigma_1^{-1} - \Sigma_2^{-1})x_0 + 0.5(\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2) + 0.5\left(ln\frac{|\Sigma_1|}{|\Sigma_2|}\right)$$

And x_0 classifies as c_2 otherwise.

To classify multiclass objects, a common approach is to create classifiers for each pair of classes and combine those classifiers to form a final classification model.

Classification Tree

We are using a binary classification tree. A classification tree is constructed by splitting the data set into two subsets based on the fastest way of decreasing the total *impurity* of the child nodes. The splitting is recursively performed until it is either non-feasible or all elements of the derived subset can be classified as a singular class.

The *impurity* of a node is measured according to the *deviance*. The impurity of a leaf i is the *deviance*

$$D_i = -2 \sum_{\text{class } c \text{ at leaf } i} n_{ic} log p_{c|i}$$

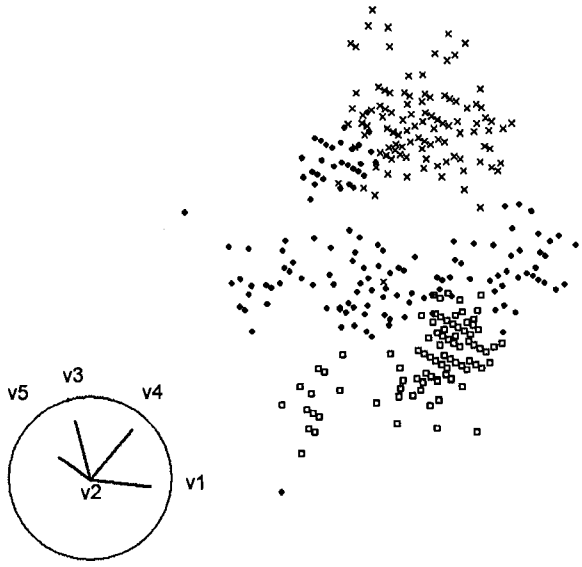
where n_{ic} is number of class c cases that reach leaf i. $p_{c|i} = \frac{p_{ic}}{p_i}$, p_{ic} is the probability that a case reaches leaf i and p_i is the probability of reaching leaf i. $p_{c|i}$ can be estimated by $\frac{n_{ic}}{n_i}$, where n_i is the total number instances at leaf i.

And the *deviance* at the node T is

$$D(T) = \sum_{\text{leaves } i \in T} D_i$$

As an example, a classification tree model is built for a training data set that consists of 500 vectors. The RSS values for each vector come from the part of our

Figure 4.1: *Plot 500 vectors that consist of three classes shown in three different markers in a two-dimensional space. Since there are five variables in each vector, we cast a five-dimensional space to a random two-dimensional space. In the picked two-dimensional space, there is no way to separate the 500 vectors.*

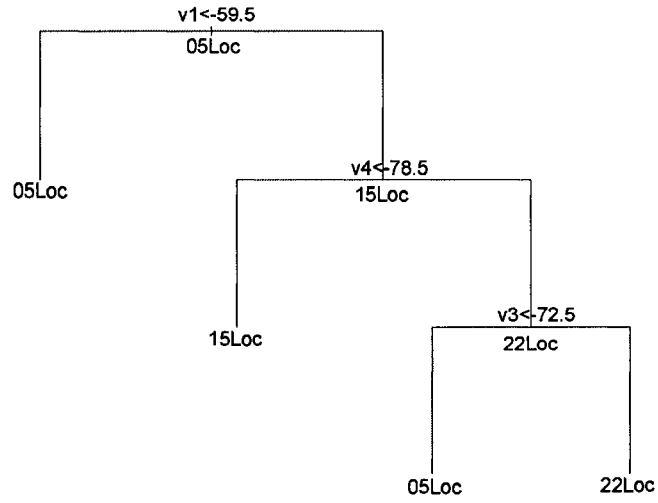


simulation data set that was generated to evaluate our proposed RSS-based NIDS (the details are shown in section 5.2). Each vector contains seven variables, $T = (v_1, v_2, v_3, v_4, v_5, i, o)$. We assume o is fixed and $i \in \{05Loc, 15Loc, 22Loc\}$ represents locations of wireless nodes. Table 4.1 shows parts of the training data.

Because o is constant and i is used as a *class label*, there are five variables in each training vector. Figure 4.1 shows a plot of five hundred vectors. A generated classification tree is shown in Figure 4.2.

According to the trained classification tree model, by giving a signalprint S, if the RSS value of v_1 is less than -59.5 dBm the signalprint is related to location 05Loc. If the RSS value of v_1 is bigger than -59.5 dBm and v_4 is less than -78.5 dBm the signalprint is related to location 15Loc. If the RSS value of v_1 is bigger than -59.5 dBm and v_4 is bigger than -78.5 dBm and v_3 is bigger than -72.5 dBm, then the signalprint is related to location 22Loc, otherwise the signalprint is also related to

Figure 4.2: *A Classification Tree.*

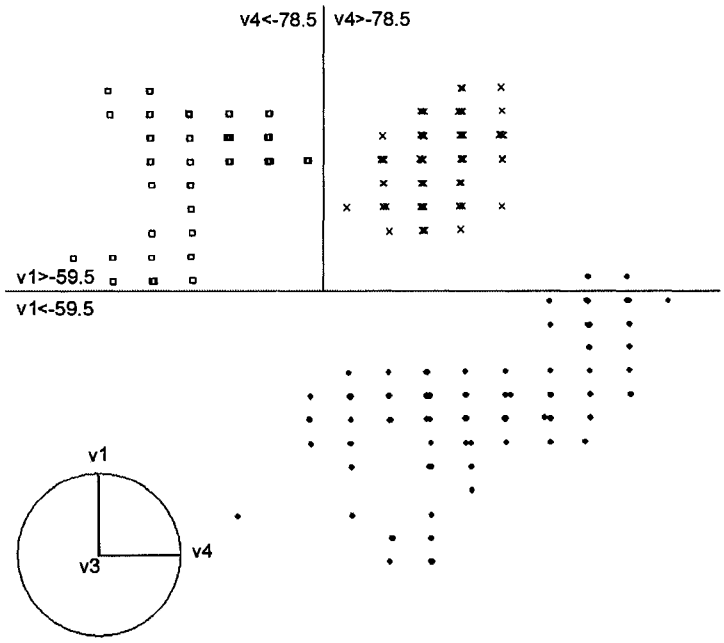


05Loc. The classification tree only splits the data set according to three variables, v_1, v_3, v_4. Figure 4.3 illustrates parts of splitting.

4.1.3 Model Selection

Based on the result of our model simulation and evaluation in Section 5.4.3, the error rate is high when using data mining models to distinguish any two neighboring network nodes with the separation distance of 1.2 meters or more. Even under the condition that the antenna orientations of all wireless devices are fixed, the error rate is 7.4 percent when using the best localization model, the QDA model, compared with the LDA (error rate is 19.5 percent) and classification tree models (error rate is 19.1 percent). The error rate is lower if we relax the required accuracy level (the separation distance of nodes) of the localization models. We use the localization models to distinguish nodes that have the separation distance of 2.4 meters or more, the error rate of the QDA model is 2.4 percent. When the separation distance is above 4.8 meters, the error rate of the best localization model, the QDA model, reaches zero. The error rate is defined as follows:

Figure 4.3: *The training vectors are cast on a two-dimensional space* $v_1 \times v4$. *The three classes of data are split by using variable* v_1 *and* v_4. *The splitting using variable* v_3 *does not show because the variable* v_3 *is in the space orthogonal to the space* $v_1 \times v4$.



$$\text{error rate} = \frac{\text{the number of misclassified signalprints}}{\text{the total number of signalprints}}$$

The LDA model has intermediate performance between the classification tree and QDA model. In the same way as the QDA model, the performance is improved greatly when we relax the required accuracy level of the localization model. When the separation distance of wireless nodes is 2.4 meters or more, the error rate is less than 1 percent. The error rate of the LDA model also achieve zero when the separation distance of any neighboring nodes is 4.8 meters or more.

The performance of a classification tree model is the worst according to our evaluation. The most important reason is that the classification tree model is *data hungry.* The model loses accuracy without enough training data. The error rate is 19.5 percent for differentiating wireless devices with the separation distance of 1.2 or more. Even for wireless devices with the separation distance of 4.8 meter or more, the error

rate is 0.6 percent, but the LDA and QDA models achieve zero error rate at such required accuracy level.

The advantage of the LDA model is that it is simple and does not need a large training data set, so that the model training is faster. However, the accuracy of the LDA model is much lower compared to the QDA model when we do not hold the assumption that all antenna orientations of wireless devices are fixed. Then the advantage of the LDA model becomes useless. The QDA model has the lowest error rate under same required model accuracy. Although it is not as fast as the LDA model, it is still the best choice when we consider more variables.

4.1.4 More Variables

If the assumption that all wireless devices have a fixed antenna orientation does not hold, the error rate becomes much worse. The training data only contains the RSS values and location information, $T = (v_1, v_2, v_3, v_4, v_5, i)$. When localizing wireless devices with the separation distance of 1.2 meters or more, the error rates of the LDA model, the QDA model and the classification tree mode are 57.1 percent, 33.4 percent and 59.3 percent respectively. When localizing clients with the separation distance of 2.4 meters or more, the error rate of any model is still bigger than 12 percent. Even the separation distance goes to 4.8 meter or more, none of the models can achieve an acceptable error rate for a NIDS. The QDA model achieves the lowest error rate, 3.8 percent.

4.2 Accuracy Enhancement

Indeed, there is a fundamental limitation in the localization of indoor wireless clients based on the RSS [9]. However, we can make the assumption that most of the attacks cannot be launched with only a very small amount of packets. A method to enhance the accuracy can be designed based on this assumption. We also provide a solution for detecting attacks that are launched by only sending a small amount of packets, such as a de-association/ de-authentication attack.

When we plot the result of locations estimated by a localization model and the actual locations in a *confusion matrix*, Table 4.2, we observe that the number of correct

Table 4.2: *A confusion matrix for the LDA model.*

	17Loc	20Loc	21Loc	22Loc	23Loc	30Loc	31Loc	32Loc	34Loc	35Loc
17Loc	92	0	0	0	0	0	0	0	0	0
20Loc	1	75	12	0	0	0	0	0	0	0
21Loc	7	27	109	13	0	0	0	0	0	0
22Loc	0	0	11	101	4	0	0	0	0	0
23Loc	0	0	0	5	114	0	0	0	0	0
30Loc	0	0	0	0	0	85	1	0	0	0
31Loc	0	0	0	0	0	0	113	16	0	1
32Loc	0	0	0	0	0	0	1	55	15	34
34Loc	0	0	0	0	0	0	2	11	98	4
35Loc	0	0	0	0	0	0	0	26	10	74

Error Rate: 18%

estimated locations for each signalprint is clustered. In the *confusion matrix*, each row represents the signalprints in a estimated location, while each column of the matrix represents the signalprints in an actual location. Table 4.2 shows the *confusion matrix* for the LDA model. The input data includes more than one thousand signalprints. The nodes are located in ten locations. The separation distance between any neighboring nodes is 2.4 meters or more. For each location, the antenna orientation of a transmitter is an unknown variable. The observation shows that although the error rate is very high, 18 percent, the correct estimated locations for each signalprint are clustered on the diagonal of the *confusion matrix*. The diagonal of a *confusion matrix* represents the correct estimation made by the LDA model.

Based on the observation, we can trade accuracy for efficiency. Suppose we accumulate multiple estimates for a specific device and decide the final estimated location using an histogram. Our evaluation shows that the performance of localization is improved dramatically with a voting in a certain number of signalprints, i.e., twenty signalprints. The detail analysis is presented in Section 5.4.5. On the other hand, a delay is added to the localization model because it has to collect enough number of signalprints.

However, how can we apply this enhancement to a RSS-based NIDS? To make the new method suitable for a RSS-based NIDS, we cannot ignore the possibility that an attacker will attack a wireless user by sending a small amount of packets. If we

only use the voting result for intrusion detection, the signalprints representing the packets sent by an attacker could be ignored because of the small number. Another observation from Table 4.2 shows that the misclassified signalprints do not fall into locations very far away from their correct location (the diagonal of a *confusion matrix*). By using this observation and the assumption that an attacker will remain a certain distance from the targeted nodes to hide its location, and the distance usually is much bigger than a localization model error, we can mitigate the weakness of the enhancement method. The localization model will not only report the most probable localization according to a group of signalprints, but report the signalprints that are shown to be abnormally far from the location where they are supposed to be. We know the correct location because in our proposed NIDS, any signalprints and their related source MAC address are linked together to form a pair, and the source MAC address is used as an identity. In a defined time window, if most of signalprints related to a specific wireless device shows that the device is in a certain location, and then the NIDS assume that location is the correct one for that sender. A threshold, E, is defined for the maximum localization error. There is a very small possibility that the distance of a misclassified location and the correct location is greater than E. A special message is triggered to raise the level of the suspicion of attacks when a signalprint is localized to be far from the correct location, and the distance is greater than the E.

4.3 A Framework of NIDS

We assume WiFi/802.11 or WiFi/802.11i networks, although we believe that our method can be helpful to other wireless network security systems. We also assume that every node has a fixed location or only moves within a given set of locations. For example, two such locations can be an office or a conference room. The protected WLAN may contain one AP or multiple APs, a so-called expanded service set (ESS). When an ESS is deployed, we assume that the multiple APs are sharing a common channel. For a multiple channel ESS, a possible solution is to alternatively monitor the different channels.

According to the attack model listed in Section 2.2, a NIDS needs to detect MAC

spoofing attacks in order to protect a WiFi/802.11 network from identity-based attacks, DoS attacks, and brute force attacks that break the WEP key and WPA keys. The NIDS can also detect rogue AP attacks since it can localize APs based on the packets they send out. The NIDS has the capability of attacker localization to help the disaster recovery after MAC layer based DoS attacks or jamming attacks have been launched.

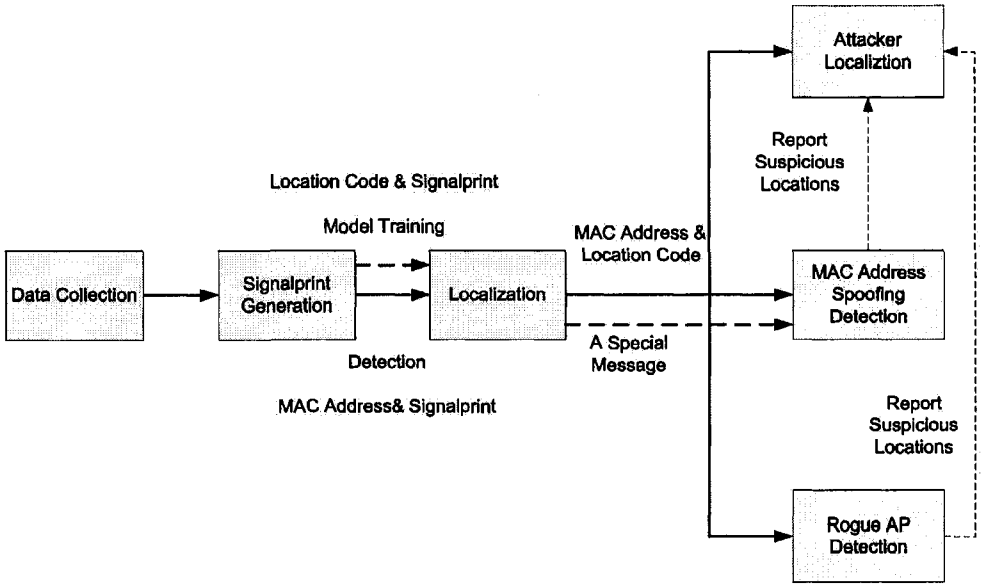
4.3.1 System Structure

The NIDS includes six parts. They are a data collection module, a signalprint generation module, a localization module, a MAC spoofing detector, a rogue AP detector, and an attacker localization module. The system structure is shown in Figure 4.4. After a localization model training mode, the signalprints and related source MAC addresses are sent to the localization module. The localization module accumulates enough data to estimate the most probable location of each node in WLAN and sends the information to the attack detectors and attack localization module. The attack detectors (MAC spoofing detector and rogue AP detector) are trained using the list of all legitimate nodes' MAC addresses and their probable locations. After the anomaly based detectors are trained, they can raise alarms when abnormal network activities are detected. Meanwhile, the locations related to a suspicious sender are reported to the attack localization module. The localization module provides assistance for locating the attackers or for recovering from a disturbed network communication.

4.3.2 Data Model

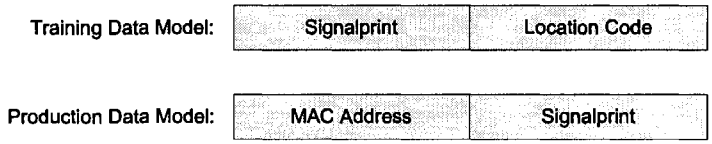
The signalprint is the most important part of the data model used by the NIDS. The data model of the proposed system is a list. For detection mode, each entry of the list pairs a source MAC address and a signalprint. During the model training mode, instead of the source MAC address, each entry of the list pairs signalprint and location code (e.g., if we define the network node location by using a Cartesian coordinate system, instead of x-coordinate, y-coordinate and z-coordinate of a location, we use a short integer number to represent them). The length of signalprint depends on the number of monitors. The location code is retrieved from the packets that are sent

Figure 4.4: *Modules in the NIDS and the working flow.*



from a training program running on every node.

Figure 4.5: *Data model for training and detecting*



4.3.3 Data Collection and Signalprint Generation

Signalprints are generated for each packet sent from a node. The length of a signalprint is fixed and corresponds to the number of monitors. An ESS may contain multiple APs in order to cover a large area. The number of monitors should be larger than the number of APs. For good performance, any spot in an ESS should be covered by at lease three monitors. An example network environment is shown in Figure 3.1.

The generation of signalprint is addressed in section 3.1.3. The data collection module extracts data, including the RSS value, source MAC address and frame sequence number, for each packet. During the system training mode, the x-coordinate, y-coordinate and z-coordinate of each related location is also collected and translated to the location code. The signalprint generation module generates the signalprint for each packet and inputs the signalprint and the related source MAC address to the localization module. The signalprint and related location code are sent to the localization module when it runs in training mode.

4.3.4 Localization Module

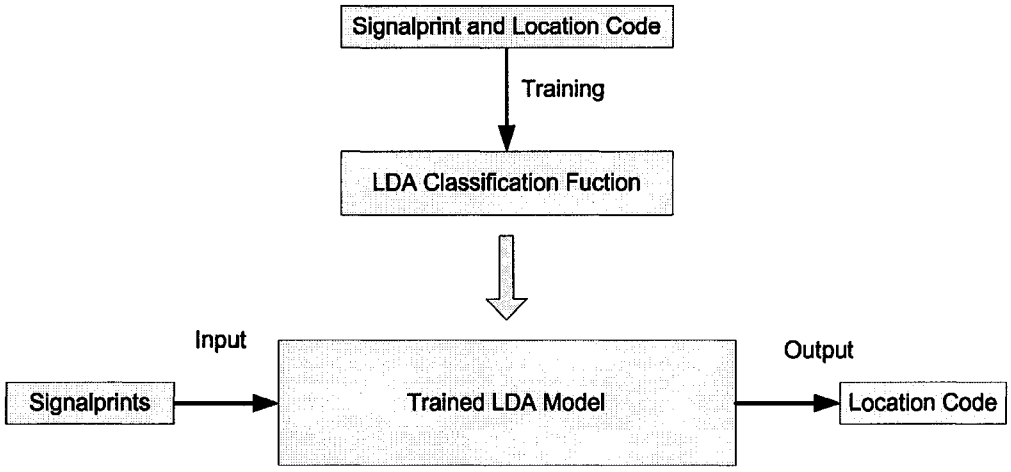
The localization module is the core of the NIDS. The localization model is based on the QDA classification algorithm. According to our experiments, the localization model that is based on the LDA classification algorithm is faster and it does not require a huge training data set. However, the accuracy is not as good as the model with the QDA classification algorithm. Especially when determining sender's location by using the enhancement algorithm based on a group of signalprints instead of single one, the QDA-based localization model can achieve a lower error rate by using a much smaller group size, therefore, the QDA-based localization model is more efficient than the LDA model. The detail analysis is addressed in Section 4.2.

Training

The training process is shown on Figure 4.6. The localization module first builds a training data set for the QDA classification model. The training data is a matrix, and each row contains signalprints and related location codes. A threshold, Q, is set up for the minimum number of signalprints related to a location. The minimum value for Q affects the accuracy of the QDA classification model. The accuracy of the QDA model is proportional to the value of Q. However, we also need to consider the efficiency of the QDA model. The Q at least is equal to five hundred. The training data set keeps growing until the number of signalprints for each location is larger than Q. When Q signalprints per each location have been acquired, the QDA-based localization model is trained. The column of *location codes* in the training data matrix is used as a *class*

label for the QDA classification algorithm. The trained localization model is able to estimate the locations of nodes using the signalprints that are generated for the packets sent from the nodes.

Figure 4.6: *The training of a localization model*



Location Detection

The result of our experiments shows that there is a fundamental limitation in accurately locating a node in an indoor WLAN environment. We propose a method to enhance the localization performance of the RSS-based localization model in Section 4.2. Using this method, the localization model estimates the sender's location using multiple signalprints.

Figure 4.7 illustrates the process of localization. The enhancement method is applied to a QDA localization model. The localization module uses queues to hold the signalprints for each different source MAC address, which indicates an individual node in a WLAN. The localization module first pushes the signalprint and source MAC address pairs into the queues when the data is received from the signalprint generation module. A threshold L is introduced to adjust the performance and efficiency of the localization model. The L is the length of each queue that is used to hold a group of signalprints related to the same source MAC address. Whenever a queue is full, the

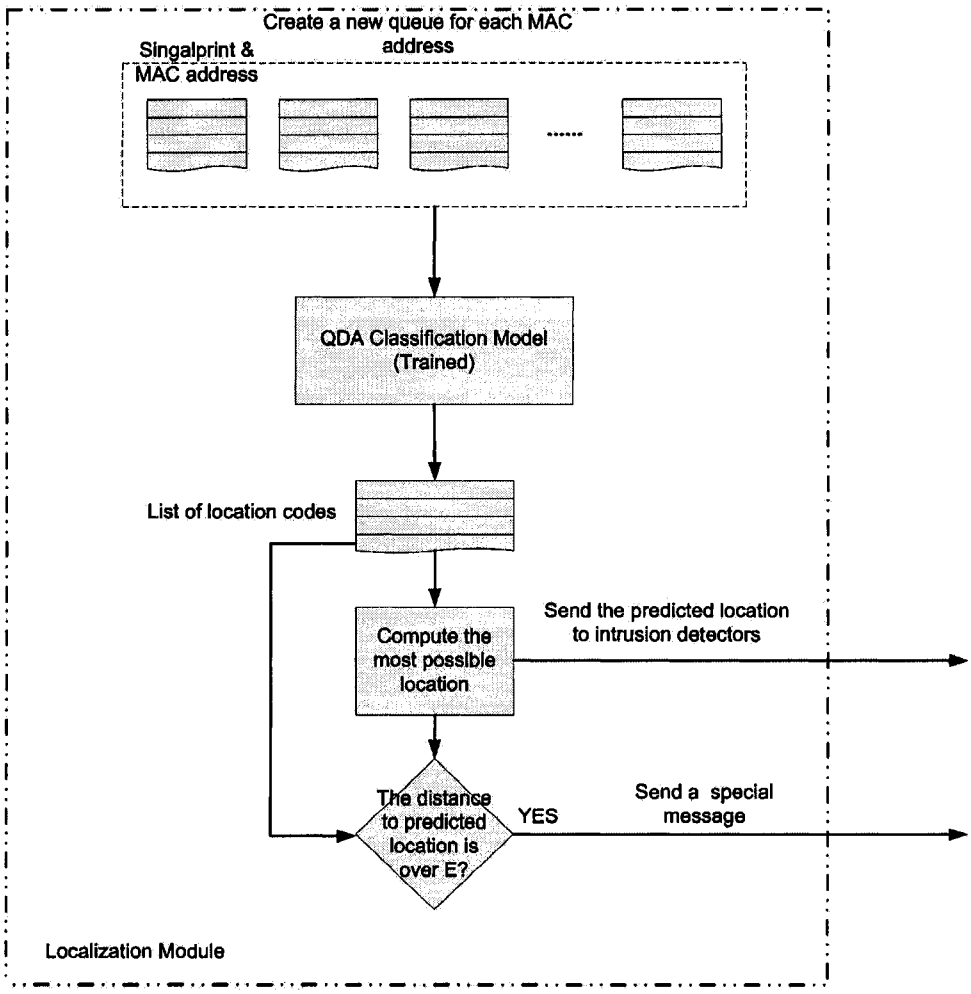
signalprints are inputted into the QDA classification model and the queue is clean. This means that the QDA localization model uses a group of signalprints instead of a single one to estimate the location of an individual node. The group size is L. The accuracy of the localization is proportional to the value of L. But if L is too large, it may degrade the sensitivity of NIDS and also may increase the false alarm rate according to the attack simulation in Section 5.5. The model outputs a result list of estimated locations. According to a percentage for each location in the list, the localization model uses the most probable location as the estimated location of a node. If more than one location in the list has an equal probability, then we have a tie. The tie is broken by randomly choosing a location among them. Meanwhile, the localization model computes the distance between each location in the list and the estimated location. If the value is over a threshold E, then a special message is sent to the MAC spoofing detector. The estimated location of the group of signalprints is also sent to the MAC spoofing detector and rogue AP detector. The special message also indicates the source MAC address of the signalprint and location. It is used to detect attacks that can be launched by sending a small number of packets.

4.3.5 MAC Spoofing Detector

Figure 4.8 shows how the list is built. The MAC spoofing detector uses an anomaly based detection model. The normal pattern is built based on a list of source MAC addresses and location codes received from the localization module. During the training mode, the detector inserts MAC addresses and location codes to a list. The MAC addresses are used as the index of the list. For each entry on the list, multiple location codes are recorded since we assume that a network node can be in one of several locations. Besides, we tolerate slight changes of location.

To detect MAC address spoofing, there are several scenarios. The first scenario is when a pair of source MAC addresses and location codes appears abnormal, i.e., the location does not exist in the entry for the node. This indicates that an attacker has spoofed a legitimate node's MAC address but the location of the attacker is not related to the node at all. A threshold, e, is introduced to tolerate localization errors. e is the maximum distance between any location in the entry for a node and

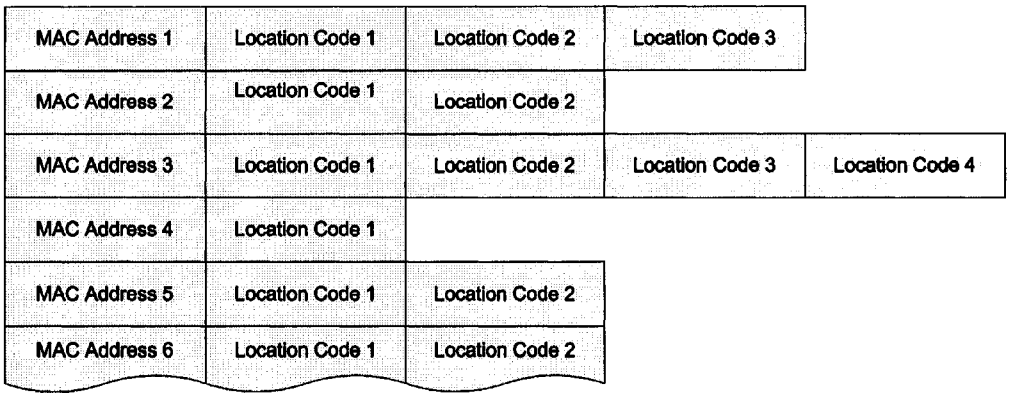
Figure 4.7: *The working process of the localization module*



a candidate location in a pair of input data. If a pair of source MAC address and location code appears abnormal and the distance between a normal location and a candidate location is larger than e, an alarm is raised and the location information and MAC address are sent to the attacker localization module.

The second scenario is when an attacker launches a MAC address attack on a location where the legitimate node could be present. In this scenario, the pairs of MAC address and location code are considered normal since the location code exists in the entry of the node. To detect this kind of attack, we assume that a legitimate

Figure 4.8: *The list forms a normal pattern.*



node is running. We set up a time window, t. t is a short time interval during which it is impossible for a legitimate node to move from one location to another location. Using a queue for each MAC address in the list, we post the pair of a MAC address and a location code that appears to be normal into the queue. The inserted pair is deleted after a time window, t. After each post, and if there is more than one pair of data in the queue, using the location codes in inserted pairs, the detector computes the maximum distance between pairs in the queue. If the distance is larger than the threshold e, then an attack is detected.

The third scenario is when an attacker launches a MAC address attack and performs another attack that needs sending a small number of packets, such as a de-association/de-authentication attack. The method for the first two scenarios cannot detect this kind of attack. The MAC address and location code sent by the location module is the most probable estimate result based on a group of signalprints. The signalprints constructed with the packets sent by an attacker are ignored because they are few in number. So we design the localization model to send the pair of source MAC address and location code to the MAC address spoofing detector, when the location is too far from the most probable location. Using this information, it is possible to detect this kind of attacks. Wherever an attacker launches the attack, we assume a relatively large distance from the location of the legitimate node to the

location of the attacker. According to our experiments, the performance of the localization model improves greatly when the distance between two nodes is relatively large, such as ten meters. This means that it is suspicious if a source MAC address is paired with a location code far from the estimated location. The detector counts the number of such pairs in a time window t_1. If the number is over a threshold p, then the detector raises an alarm.

4.3.6 Rogue AP Detector

The rogue AP detector is also based on anomaly detection model. The normal pattern is built on a list containing the source MAC addresses of the authorized APs and the location codes of those APs. Each MAC address may relate to several locations because of the error rate of the localization model usually caused by network environment changes. Before the model training starts, the MAC addresses of all authorized AP are recorded into a list and the MAC address is used as the index of the list. During model training, after pairs of source MAC address and location code are received from the localization module, all location codes related to any authorized AP are added into the list. A threshold, N, is set up to limit the maximum length of each entry of the list, i.e. the maximum number of locations of an AP. The bigger N can make the detector tolerate the localization error and decrease the false alarm rate. However, the bigger N will slightly decrease the sensitive of the detector.

During the detection mode, whenever the detector finds an AP with a MAC address not contained in the MAC address list of authorized APs, the detector raises an alarm. Meanwhile, the detector pinpoints the location of the unauthorized AP using the signalprints of the packets sent from that AP, and sends the location and MAC address of suspicious APs to the attacker localization module. On the other hand, whenever the detector finds that a MAC address and location code pair appears to be abnormal, the detector raises an alarm and pinpoints the location of the suspicious AP. A pair is considered abnormal when the location code is not contained in the location code list related to the MAC address in the pair. A distance threshold e can also be set up to tolerate the localization error and decrease the false alarm rate.

4.3.7 Attack Localization and Disaster Recovery

The localization capability of the NIDS can be used to locate attackers and to help disaster recovery after a MAC layer related DoS attack or a jamming attack. Whenever the MAC spoofing detector and rogue AP detector find suspicious activities, they not only raise alarms but also send the locations of the suspicious sender to the attacker localization module. The information provided by that module is used to pinpoint the location of the suspicious sender. On the other hand, after an attack launched DoS attack or a Jamming attack that exploits WiFi/802.11 MAC layer vulnerabilities, even when the attack is launched by a legitimate user, the localization module can pinpoint the location of the sender that caused the DoS attack or network jamming. The location information can help to rapidly restore the network operation.

Chapter 5

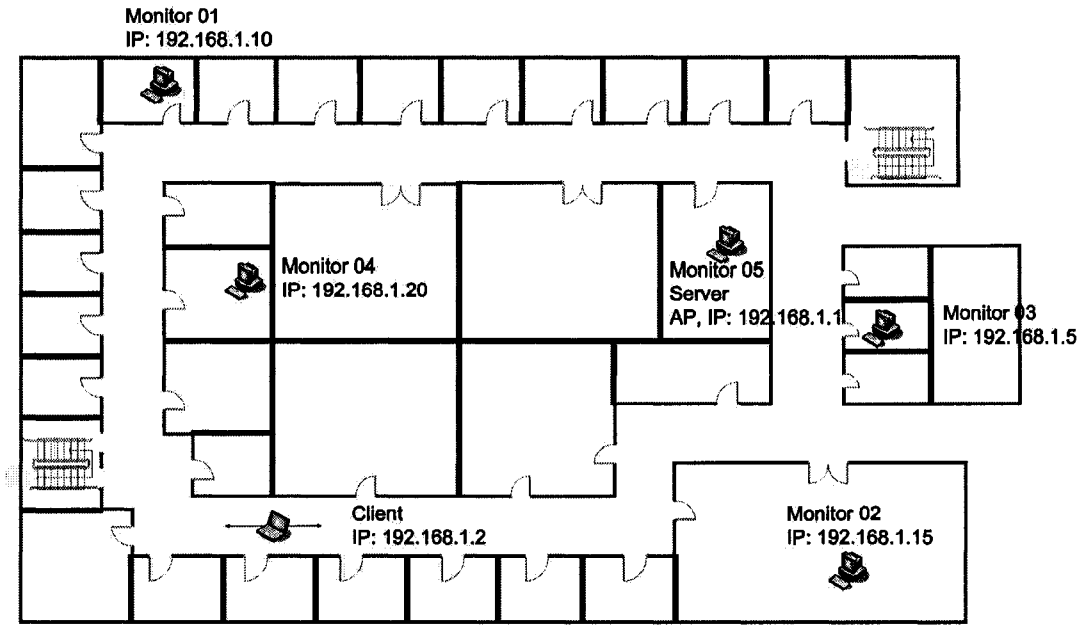
Evaluation of the Enhanced Signalprint Method

Experiments are conducted to analyze the feasibility of applying data mining algorithms in an RSS-based localization for an indoor WLAN environment and evaluate the improvement of applying new methods to an RSS-based NIDS for WiFi/802.11 networks. Using a test-bed, we collected raw frames sent out from 59 locations in order to generate a data set containing 47,200 signalprints for evaluating localization models and attack detectors. Based on the evaluation results, we proposed and tested an enhanced method. We also designed a MAC address spoofing detector using the matching rule algorithm proposed by Faria and Cheriton [10] to compare the matching rule algorithm with the proposed NIDS. The comparison between a data mining based attack detector and a signalprint-matching-based attack detector is based on a simulation of MAC address spoofing attacks. We also evaluated an attack detector using the enhanced method.

5.1 Test-bed Setup

Our experimental test-bed was set up on the 5^{th} floor of Herzberg building in Carleton University. As shown in Figure 5.1, the test-bed included one wireless client, five wireless monitors and one server. All monitors were located in different rooms. The monitor 05 was also set up as a server and AP. A wireless client moved along the hallway and sent 800 packets to AP on 59 pre-defined positions one by one. The 800 packets were divided into four parts. Every part contained 200 packets that were sent when the antenna of the wireless client faces different directions. We used four directions: East, South, West, and North. The distances between positions were 1.2 or 2.4 meters.

Figure 5.1: *The setup of the experiment. Five monitors and one server are set up in different rooms. A wireless client moves along the hallway.*



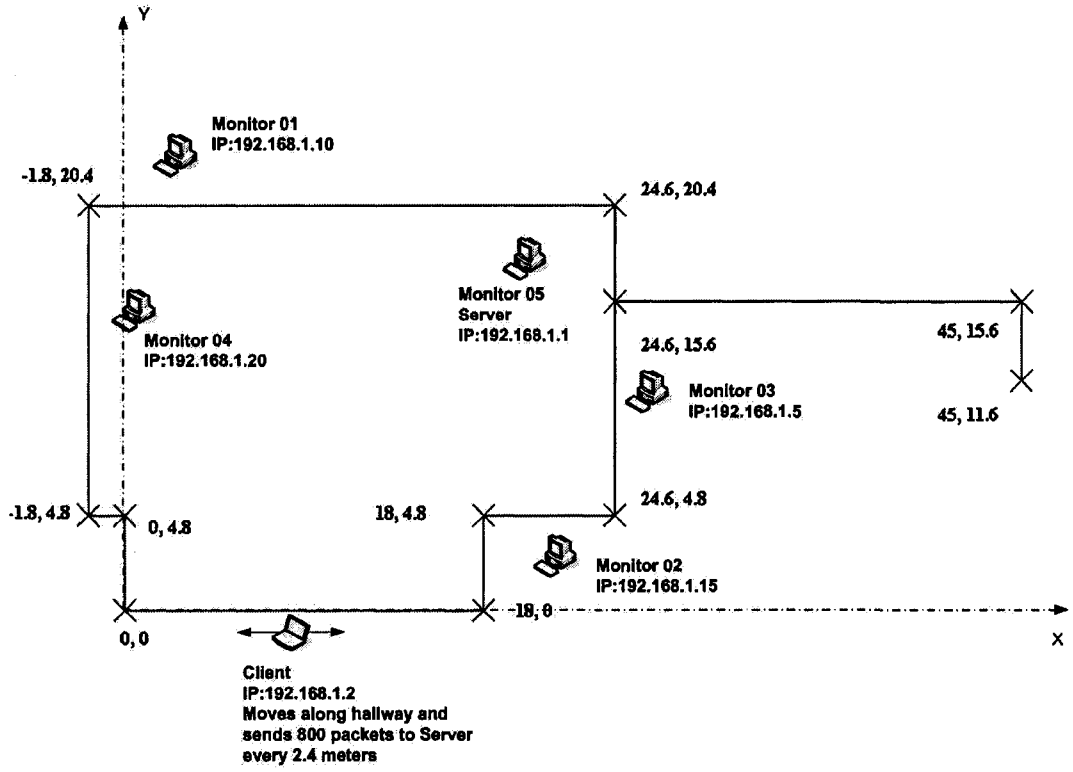
5.1.1 Hardware

In our experiment, there was only one laptop moving along the hallway and sending the packets to the server at 59 different positions. Actually, if the localization models can successfully differentiate the packets sent from those 59 locations, they also can differentiate multiple wireless clients that are located in those 59 different positions. This is because the diversity of network adapters can only improve the performance of data mining models.

All five monitors were set up on desktop computers. We used TRENDnet Wireless PCI adapter, which is based on Atheros 5212 chipset. An open source wireless adapter driver for Linux, MADWiFi [22], is available to help to set up the wireless adapter in monitor mode. Instead of *prism2 header*, we configured the monitors to capture the *radiotap header*, such that we can extract the RSS value for each captured packet from it.

The AP used an enhanced antenna to make a single AP covering a bigger area.

Figure 5.2: *Geometry of device location.*



The antenna gains 8 dBm sensitivity. The desktop that was configured as AP also plays roles as server and monitor. The server was used to remotely manage monitors and collect raw frame data generated by monitors. The wireless client was a laptop with a USB interface wireless adapter

5.1.2 Monitors

All monitors ran Fedora core 6, because the open source wireless adapter driver is handy and satisfies our requirements. To capture and dump the raw frames monitored by MADWiFi driver, an open source network protocol analyzer, WireShark [33], was deployed. WireShark supports multi-platform and hundreds of protocols. It also provides powerful filters to help us decrease the size of raw frame data by only dumping the frames we need.

5.1.3 Send and Receive Packets

We developed a tool called SAR to create experimental packet streams. The SAR performs the operations of sending and receiving packets. It ran under the sending mode on the client and the receiving mode on the server. The SAR on the client sent pre-defined messages to the SAR on the server through a TCP/IP protocol. The message contained useful information, such as a serial number, antenna orientation of the sender, and so on. That information will be used to simplify the data set generation.

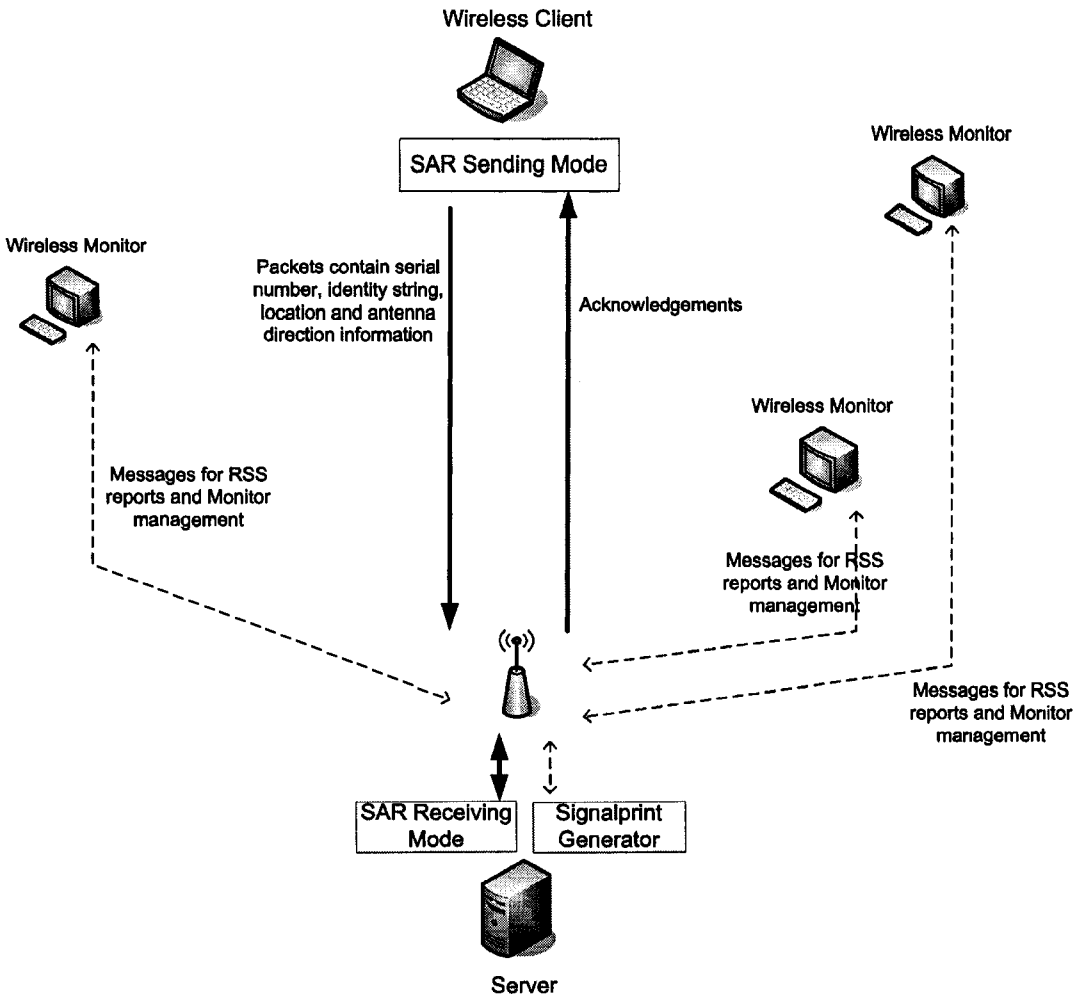
Figure 5.3 illustrates the operations of SAR. First of all, the server side SAR opens a port for the incoming messages, and then the client side SAR starts sending messages to it after all parameters are set up. The parameters included the port number, which was open on server, the x-coordinate, the y-coordinate and z-coordinate of a location, the direction of the laptop's antenna and the number of packets sent on each position.

To simplify the signalprint generation, instead of using frame sequence numbers, we set up a serial number for each packet. A serial number was added to each message sent by SAR. It was only used to differentiate packets that are sent on the same position. The location information and antenna orientation were able to differentiate the packets sent from different locations and directions. According to those serial numbers and location information, the signalprint generator could identify the RSS values related to each packet that reported from different monitors, and then combine the RSS values to the signalprint for each packet.

Another simplification was designed for WireShark filtering. SAR adds an unique string in every packet in order to utilize the filters in WireShark to dump the packet sent by the client. For example, the frames are related to the communication between monitors and the server were not captured.

The server side SAR responded to the client by sending acknowledgements for each received message. The acknowledged message could inform the wireless client that all packets were received, so that the wireless client can move to the next location or change its antenna direction. Moreover, the acknowledged message could help the client to find out whether a location is out of the range covered by the AP.

Figure 5.3: *The SAR operation.*



5.1.4 Geometry

The experiment was deployed on part of the 5^{th} floor of Herzberg building. Because only one AP was set up, the signal could not go further than 25 meters from the AP. Shown on Figure 5.2, a Cartesian coordinate system is used to define the nodes' location. The origin point was set on the Southwest corner. The server was located in a library (Monitor 05). The other four monitors were setup in four rooms separately.

The width of the hallway is 1.8 meters. The distance between two neighboring

Table 5.1: *Part of a data mining data set. The Location column contains integer numbers (location code) that represent the actual x-coordinate, y-coordinate, and z-coordinate of a location. We also define integers* $0, 1, 2, 3$ *to represent the four different antenna directions.*

Location	RSS 1	RSS 2	RSS 3	RSS 4	RSS 5	Orientation
1	-67	-59	-73	-70	-73	1
2	-53	-52	-72	-80	-69	2
3	-56	-60	-100	-100	-72	1

observation positions was either 1.2 or 2.4 meters. According to research, the accuracy of a RSS-based localization model is unlikely to reach 1.2 meters with an acceptable error rate. The 1.2 meters distance is applied between neighboring observations on the West hallway. Others use the 2.4 meters distance.

5.2 Build a Data Set

Using the raw frame dumps collected by monitors, a data set for a data mining model and a signalprint-matching model was created. The data set was not only used to choose the best data mining algorithm among LDA, QDA, and the classification tree, but also to perform a MAC address attack simulation for a comparison between data mining based detectors and a signalprint-matching based detector.

Table 5.1 gives an example of part of a data set. The data set is a matrix, and the columns of the matrix contain variables that can be used by localization models or attack detectors, such as the location of client, RSS values from different monitors, and so on. Each row of the matrix represents an observation for an individual packet. The values listed in the same row are information about the same packet. From the point of view of the signalprint concept, each row contains the signalprint for an observed packet, the location code of that packet, and the antenna orientation information of the node that sent that packet.

5.2.1 Data Collection

Starting from the origin point, the client moved along the hallway and sent packets on the 59 pre-defined positions one by one. On each position, the antenna direction changed four times. 200 packets were sent out to the same direction and position. The monitors captured the packets sent from the client and dump the raw frame data onto its local hard disk. Later, the server collected all the data from each monitor.

A tool was developed to convert the raw frame data to the data set we need. First of all, the individual frames dumped by the same monitor were separated and the RSS value and other information for each packet are extracted. Then, according to the location and serial number, the RSS values and other information related to the same packet were combined to form a row in the data set. Finally, all information for each packet was inserted into the data set. We use MySQL database to manage the data set.

In the data set, we used a location code to represent the actual x-coordinate, y-coordinate and z-coordinate of a location. As shown in Figure 5.2, we set the Southwest corner as the origin of a three-dimensional Cartesian coordinate system. In our experiment, the z-coordinate is always set as zero. Instead using the x-coordinate, y-coordinate and z-coordinate of a location, we used the integer numbers one to 59 as location codes to represent the 59 pre-defined locations. We also use four integer numbers to represent the four different antenna directions.

5.3 The RSS Variation

The RSS oscillation is one of the most important reasons that cause the inaccuracy of the RSS-based localization model. We first ploted RSS values of packets sent from three different locations to show the RSS oscillation. The RSS values were extracted from the same monitor. Figure 5.4 shows the plot of RSS values and histograms. Figure 5.4 a) uses the data that the antenna of the laptop faces East when packets are sent. Figure 5.4 b) uses the data that we assume that the antenna direction is unknown when the packets are sent. The number of packets for a location is 200 and 800 respectively.

The histograms in Figure 5.4 b) show that there are about 8 dBm RSS oscillations when the orientation of the wireless client is an unknown variable. When we assume the orientations of antenna are fixed, the RSS oscillation is generally degraded, but strong RSS oscillation sometimes occurs. For example, the Location 10 in Figure 5.4 a).

5.4 Simulation and Evaluation of Data Mining Algorithms

We simulated and evaluated localization models based on the LDA, QDA, and classification tree algorithms to find a suitable algorithm for distinguishing wireless network nodes based on signalprints of the packets sent from them. For each model, two types of data set were applied, the data with the information of antenna orientation of nodes and the data without that information. The latter type of data was based on the assumption that the direction of the wireless client is an unknown variable.

5.4.1 Model Building

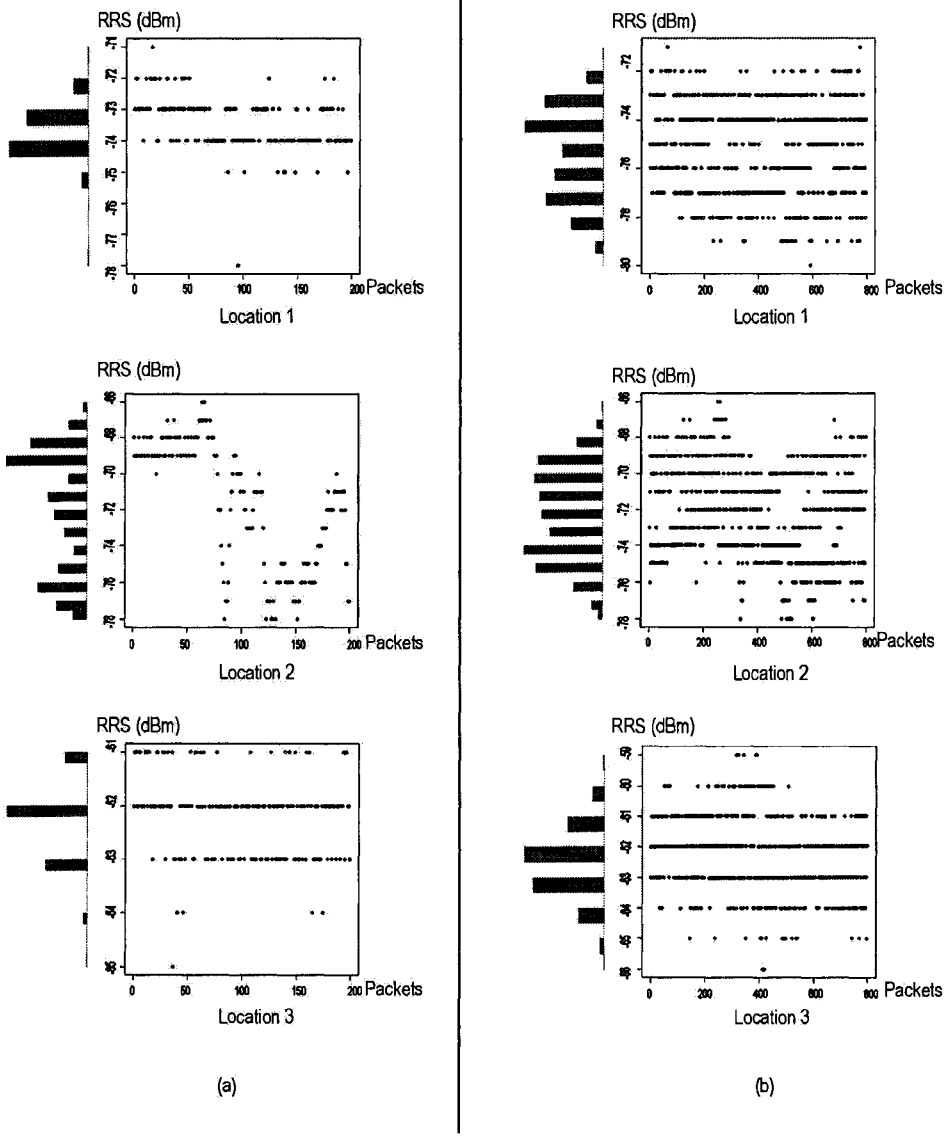
The model implementation and evaluation were based on the R programming language [25]. R is a programming language and software environment for statistical computing. Not only the LDA, QDA, and classification tree are implemented in R, but the evaluation procedure, data set importing and data set reorganizing are also designed and implemented in R. The evaluation procedure first generated a *confusion matrix* for the original location for each signalprint in the data set and the estimated location for each signalprint was output from localization model. Then the evaluation procedure calculated the estimate result based on the *confusion matrix*.

5.4.2 Data Set for Data Mining Models

Training and Testing Data

There were 59 locations and 800 packets per location. The total number of signalprints in the data set was 47,200. For the evaluation of data mining models, the data set was divided into two parts: training data and testing data. We randomly picked up 80 percent data to form a training data set. The rest of the data was used as testing

Figure 5.4: *A plot of RSS values generated by one monitor. The histogram shows the distribution of the RSS values*



data set. The training data was used to train the data mining models and the testing data was used to evaluate the accuracy of the models.

Two Scenarios

Two scenarios were concerted when we analyzed the performance of different models. According to the research of Bahl et al. [2], the RSS at a given location varies significantly depending on the antenna orientation. To test how much the antenna orientation affects the detection result, we adjusted the data set to simulate the two different scenarios. In the first scenario, the data set contains all the information including the antenna orientation. This means the antenna orientation is known and fixed. In the second scenario, the data set drops the information of the antenna orientations. This indicates that the antenna orientation is an unknown variable.

For the data mining models (LDA model, QDA model, and classification tree model), the model training was based on the training data set. To test the influence caused by antenna orientation, the models were trained for two different scenarios separately. The testing data set was used to measure the performance of different models and was also adjusted for two scenarios. In order to conduct a fairly comparable result, the models used the identical training data set and testing dataset.

Data Set Reorganization

To evaluate localization models in different levels, three different data sets were generated by grouping data according to the distance of nodes. As mentioned previously, we assumed the data was generated by 59 nodes from the 59 different locations, although the original data was generated by a single node. The distances were 1.2, 2.4, and 4.8 meters. For example, we picked up all the signalprints that the distance of the nodes related to them is 1.2 meters to form data set 1-2. The data set 2-4 and data set 4-8 are generated using the same method.

Using the reorganized data set, we could fairly evaluate the localization models on three different accuracy levels. Since for each data set, the distance of the nodes was constant, the evaluation results were not affected by the component of the data set. Meanwhile, the evaluation result was for the worst case of each localization model.

Table 5.2: *The performance of LDA, QDA and classification tree model in scenario one.*

	1.2 m	2.4 m	4.8 m
LDA	19.5%	4.4%	0%
QDA	7.4%	2.4%	0%
classification tree	19.1%	10.9%	0.6%

5.4.3 Results and Comparison

Scenario One

In this scenario, we assumed the antenna orientations of every network nodes were fixed and known. This meant that the training and testing data matrix included all the columns. In practice, this assumption may not be easy to hold since a note may change its direction from time to time.

Table 5.2 lists the error rate of LDA-, QDA-, and classification tree-based localization model when the models are used to differentiate the wireless nodes with a separation distance of 1.2 meters, 2.4 meters, and 4.8 meters. The error rate listed in the table is the avenge value on five times independent evaluations. The training and testing data sets for each evaluation are different. The definition of error rate is:

$$\text{error rate} = \frac{\text{the number of misclassified signalprints}}{\text{the total number of signalprints}}$$

When evaluating the performance of a model, we used three data sets corresponding to the different separation distance between any neighboring nodes in order to evaluate the performance for different accuracy levels. For example, there are 21 locations in all 59 pre-defined locations that any neighboring location among them has 1.2 meters distance. The signalprints related to those 21 locations were used to evaluate the performance of three models when the localization accuracy level is 1.2 meters.

The result shows that the QDA-based localization model has the best performance. Even with the dataset that the separation distances between any of two senders are 1.2 meters, the error rate is 7.4 percent. The error rate is 19.5 percent and 19.1

Table 5.3: *The performance of LDA, QDA and classification tree model in scenario two.*

	1.2 m	2.4 m	4.8 m
LDA	57.1%	30.9%	14.3%
QDA	33.4%	12.5%	3.8%
classification tree	59.3%	43%	14.9%

percent for the LDA- and classification tree-based models respectively. The error rate is proportional to the distance between nodes.

The performance of the LDA-based model improves rapidly compared with the classification tree model as we trade off some accuracy of localization. The error rate of the LDA- and QDA-based models are both zero when the localization models distinguish the network nodes so that any neighboring nodes have the separation distance of 4.8 meters. Another observation is that the LDA- and QDA-based models are much faster than classification tree-based model. The LDA-based model is slightly faster than QDA-based model. The efficiency is another important factor we need to take into account when choosing the proper localization model for NIDS.

Scenario Two

This scenario is more practical. We assumed that the antenna orientation of each network node was unknown. In other words, we allowed the senders to change their direction any time they wanted. The performance of three data mining models is shown in Table 5.3. Similar to the evaluation of scenario one, the error rates of the LDA-, QDA- and classification tree-based localization models is derived when the three data sets are used. The distances between the nodes in the data sets are 1.2 meters, 2.4 meters and 4.8 meters respectively. The error rate listed in the table is the avenge value based on five independent evaluations.

When we considered the rotation of wireless nodes, the performance of localization models degraded a lot. The lowest error rate was 3.8 percent that was achieved by QDA-based model as the separation distance between nodes was 4.8 meters. The

error rate was 14.3 percent and 14.9 percent for the LDA- and classification tree-based models, respectively. If the separation distance between nodes was 1.2 meters, all models had an extremely high error rate. The LDA model had a 57.1 percent error rate, the QDA had a 33.4 percent error rate, and classification tree had a 59.3 percent error rate.

According to the evaluation results, the QDA-based localization model always has the lowest error rate. The classification tree-based model has the worst performance (highest error rate and much slower than LDA and QDA model). Basically, the error rates are too high to build a detector for NIDS using any of those three localization models. However, based on the observations on the *confusion matrix* table in Table 5.4, that was generated by evaluation procedure when we evaluated the QDA-based model, we can enhance the performance of the localization model by using a group of signalprints instead a single one.

5.4.4 A Enhancement Method

The *confusion matrix*, Table 5.4, is based on the evaluation result of applying the QDA model on the data set when the separation distance between any neighboring nodes was 2.4 meters. The performance of the localization model was not very good, since in the scenario the antenna orientation of a node is unknown. The error rate was 10.8 percent.

In the *confusion matrix*, each row of the matrix represents the instances in an estimated location, while each column represents the instances in an actual location. The diagonal of a *confusion matrix* represents the correct estimation made by a localization model. In Table 5.4, for example, the row names and column names on the top and the left indicate a total of 22 locations in the data set of 2,534 signalprints. The *Row Sum* listed on the right of the table shows the total number of signalprints of packets sent from each location that are estimated by the localization model. The *Column Sum* on the bottom of the table shows the actual number of signalprints of the packets sent from each location. The first row of data in the table shows that in 129 signalprints that are estimated to relate to location two, 114 of them are correctly estimated by the localization model. But six of the signalprints are classified

Table 5.4: *A confusion matrix built on the estimation of the QDA localization model and the actual location of each signalprint*

	true																										
object	2	4	6	9	11	13	15	17	20	21	22	23	30	31	32	34	35	37	38	39	40	42	Row Sum
2	114	6	6	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129
4	1	111	18	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	134
6	1	9	114	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	125
9	7	3	5	145	13	3	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	179
11	1	0	0	7	102	0	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	114
13	0	0	0	2	0	76	24	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	108
15	0	0	0	0	0	21	64	16	0	0	2	0	0	0	0	0	0	0	0	0	0	0	103
17	0	0	2	3	4	3	8	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95
20	0	0	0	0	0	0	0	0	93	14	0	0	0	0	0	0	0	0	0	0	0	0	107
21	0	0	0	0	0	0	0	0	6	123	3	0	0	0	0	0	0	0	0	0	0	0	132
22	0	0	0	0	0	5	0	0	0	0	122	0	0	0	0	0	0	0	0	0	0	0	127
23	0	0	0	0	0	0	0	0	0	0	0	122	0	0	0	0	0	0	0	0	0	0	122
30	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	80
31	0	0	0	0	0	0	0	0	0	0	0	0	0	114	10	0	0	0	0	0	0	0	124
32	0	0	0	0	0	0	0	0	0	0	0	0	0	3	88	1	6	0	0	0	0	0	98
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	112	6	0	0	0	0	0	123
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	105	0	0	0	0	0	110
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	0	44
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	71	0	0	2	73
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	0	4	137
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	114	5	121
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	3	138	149
Col Sum	124	129	145	164	120	108	97	102	99	137	128	122	80	117	108	113	117	45	72	141	117	149	2534

error 0.1081294

to relate to location four. Six of the signalprints are misclassified to relate to location six, two of the signalprints are misclassified to relate to location nine, and another one are misclassified to relate to location 11. On the other hand, the first column shows us that out of 124 signalprints that are actually related to location two, 114 are correctly classified. But one of the signalprints is classified to relate to location four, which is actually related to location two, one of the signalprints is classified to relate to location six, seven of the signalprints are classified to relate to location nine, and another signalprint is misclassified to relate to location 11. All of them are actually related to location two.

One of the observations shows that although the error rate is 10.8 percent, the correct estimations for each location cluster on the diagonal of the *confusion matrix*. According to the definition of *confusion matrix*, the diagonal of a *confusion matrix* represents the correct localization estimated by a localization model. To decrease the error rate, we propose an enhancement approach by making a trade off between error rate and efficiency. Suppose we accumulate multiple localization results for multiple signalprints that are supposed to send from the same sender, and decide the final predication using a histogram. This means that we can improve the performance of localization by voting in a certain number of signalprints, although a delay may occur while the estimator is collecting enough number of signalprints.

We can use another observation to mitigate the possibility that an attacker exploits our enhancement method when the method is utilized by an NIDS. A weakness exists because the enhancement method only uses the voting result based on a group of signalprints and the size of the group could be larger than 20. The signalprints representing the packets sent by an attacker could be ignored because it is so small. We cannot ignore that an attacker may also launch attacks by sending a small amount of packets. Table 5.4 shows that the misclassified signalprints are not related to the location that is far from the correct location. Usually, the distance between the correct location and the misclassified location is less than ten meters. If we assume that an attacker will remain a certain distance from the target node in order to hide its location, this observation can be utilized to accomplish the enhancement method. The detail is represented in Section 4.2. The localization model will not only report the

estimated localization according to a group of signalprints, but report the signalprints that are shown abnormally far from the location where they are supposed to be.

5.4.5 Enhancement Result

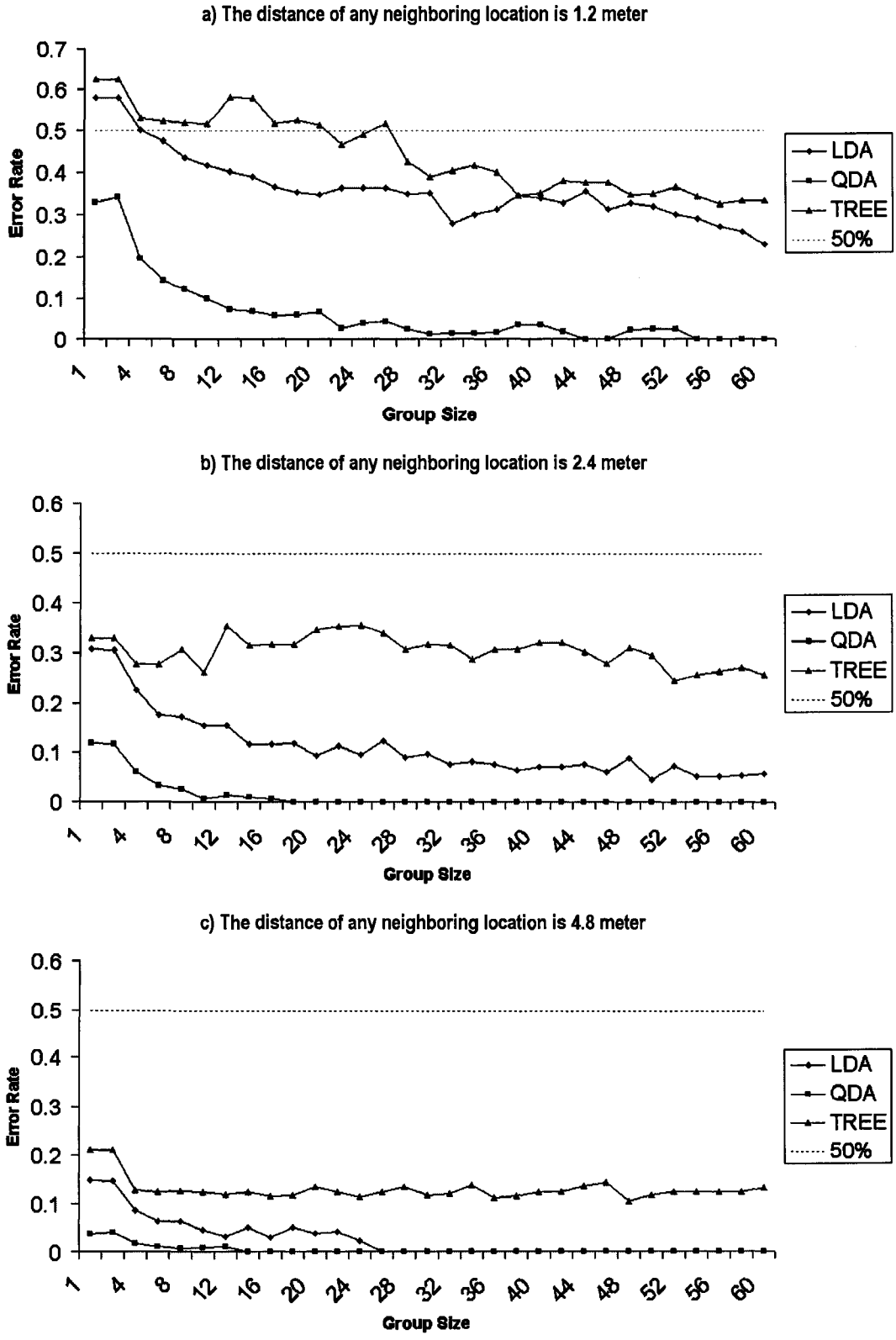
Figure 5.5 shows the comparison to apply the enhancement method on LDA, QDA, and classification tree-based localization models. Figure 5.5 a) illustrates the error rate of localization models based on the data set when the separation distance of any two neighboring nodes is 1.2 meters. Figure 5.5 b) illustrates the error rate of localization models based on the data set that the separation distance of any neighboring nodes is 2.4 meters. Figure 5.5 c) shows the error rate of localization models using the data set that the separation distance is 4.8 meters.

The error rate degrades rapidly in LDA- and QDA-based localization models, especially the QDA-based model. But the classification tree-based model does not. The reason is the insufficiency of training data. The classification tree algorithm is more *data hungry* to perform model training. If the size of training data is not large enough, the classification tree-based model cannot classify all locations. That is why the error rate stops decreasing when a certain level of error rate is reached.

When the accuracy level of the localization model is to distinguish nodes that have 1.2 meters separation distance, the QDA model achieves 1 percent error rate when the size of the group is larger than 30 and a zero error rate when the group size goes to 44. If we relax the accuracy level of localization model to distinguish nodes that have 2.4 meters separation distance, The QDA model can achieve a zero error rate when the group size is larger than 18. When the accuracy level of the localization model is further relaxed to 4.8 meters, LDA can also achieve a zero error rate. The LDA model achieves a zero error rates when the group size goes to 26; the QDA model achieves a zero error rate when the group size is larger than 14.

The QDA-based localization model is the best model to apply our enhancement approach because it can always reach zero error rates when a group of signalprints are used to localize a wireless node. Although the LDA model is a little faster than the QDA model, more signalprints are required to achieve the same error rate as QDA. So, the QDA model is recommended to use on a NIDS.

Figure 5.5: *The result of the enhancement algorithm*



5.5 Data Mining vs. Signalprint Matching

A simulation was deployed for testing the performance of QDA-based NIDS. The error rate of a localization model and the performance of attack detectors are two possible factors influencing the performance of the NIDS that are introduced in Section 4.3. The error rate of a localization model is the most important factor, because it is the basis of the NIDS. The feasibility of the NIDS is decided based on this error rate. Therefore, the simulation was focused on the evaluation of the localization model. Instead of an error rate, the Receiver Operating Characteristic (ROC) curve is used to evaluate the performance of the NIDS. As an comparison, we also designed an intrusion detector based on the concept of signalprint max-match and min-match that were introduced by Faria and Cheriton [10].

5.5.1 Attack Simulation

A typical simple MAC address attack can be easily simulated by using the data set that we have generated. As mentioned previously, we choose the MAC address attack because the MAC address attack is the most important attack that the proposed NIDS is meant to detect. The successful detection of a MAC address spoofing attack can prevent the nodes from further attacks. Moreover, the rogue AP detector in our NIDS is based on a similar method.

We assume an attacker captured the SSID of the authorized AP and cracked the WEP key by exploiting the WEP vulnerability of a WiFi/802.11 network, or cracked the WAP key of WiFi/802.11i network. As discussed previously, a WEP key and weak WAP key is easy to be cracked. During or after the attacker gain the access to the WiFi network, the MAC address spoofing attack has to be launched, otherwise the attacker is easily detected by an authorized MAC address list. A MAC address spoofing attack can be launched in two scenarios: the attack is targeted to one victim or multiple victims. In the first scenario, the attacker may wish to hide its presence, and launch a session hijacking or other identity-based attacks. In the second scenario, a DoS attack may be launched.

Attack Dataset

The data set consists of signalprints related to all 59 locations, or we can assume that there are 59 different nodes in each location. An attacker may impersonate any of those nodes. To simplify the simulation, we assumed the attacker was only located in the 59 locations where we have data. This assumption is reasonable because the NIDS can easily detect the attacker if its location is not in the normal list of the MAC address detector. The MAC address attack can be simulated by assuming a victim node located in one location with attacks launched in another different location. Therefore, the signalprints related to the location where attacks are launched are used to simulate attack activities. For example, we chose the victim node in the location of A and its location code is 5, and we assume the location of attacker is B where the location code is 18. Then the signalprints related to the packets sent from location B were treated as attack.

Since the MAC address is clear for each packet and every legitimate node (MAC address) has its normal location listed in a trained attack detector, the detectors will treat the network activities as suspicious if the packets are sent from a location that is not in the location list. If the normal list for each legitimate node is accurate, then what we need to evaluate first is the accuracy and error rate of the localization model.

To fairly evaluate the performance of the data mining based detector and signalprint-matching-based detector, we assumed that the attacks were launched from different locations and the evaluation was based on the average performance. This meant that we generated a group of attack simulation data sets for several different locations instead of only one. Each data set not only consisted of the signalprints representing attacks (positive), but contained the signalprints representing the normal network activities (negative). The ratio between positive and negative was $1 : 1$. The positive and negative signalprints were chosen from the signalprints related to the attack location and the location of normal nodes.

When we generated the attack data set for a specific location, in order to add signalprints related to normal network traffic in the data set, a few neighboring locations were chosen to simulate the normal nodes. Our experiment shows that the choice of simulated normal nodes greatly affects the results of a detector evolution.

The distance between the locations of a normal node and an attacker is proportional to the performance of detectors. We defined a threshold, a, as the minimum distance between the attack location and the chosen locations of normal nodes. The a indicated the level to which we evaluated the detector. It actually represented how close an attacker was to a victim node in order to avoid detection. We evaluated the detectors while $a = 2.4$ and $a = 4.8$ meters.

Considering the worst situation, we always chose the location representing normal nodes as nearly as possible to the attack location but further than a when picking negative signalprints. For example, the location code of an attack location was 12, and we used $a = 2.4$ meters. The simulated attack data set consisted of 500 positive signalprints related to location 12, 250 negative signalprints related to location 10, and 250 negative signalprints related to location 14, because locations 10 and 14 were nearest locations to location 12 and the distance between them was bigger or equal to 2.4 meters.

5.5.2 Evaluation Method

The MAC address detector in the NIDS proposed in Section 4.3 is based on the location information estimated by the localization model. Therefore, the performance of the NIDS depends on the localization model and attack detectors (MAC address spoofing detector and rogue AP detector). The accuracy and error rate of a localization model are the most important factors because the localization model is the basis of the NIDS. We evaluate the performance of NIDS related to the localization model and leave the evaluation of detectors to feature studies.

A ROC curve is a classical method for determining possible optimal models. A classification model or detector is a mapping of instances into a certain group. For a two-class problem, the detection result can be positive or negative. A comparison of the detection result with the actual value shows four possible outcomes: true positive (TP), false positive (FP), true negative, and false negative. The ROC analysis is based on the true positive rate (TPR) and false positive rate (FPR). TPR determines a detector performance in detecting positive instances correctly among all positive samples. FPR determines how many incorrect positive results are actually negative

among all negative samples. In our case, we supposed that the attack data set consisted of 100 signalprints representing malicious network activities (positive) and 100 signalprints representing normal network traffic (negative). A detector reported that 110 signalprints were suspicious. Compared with the actual value, there were 90 TP and 20 FP in 110 suspicious signalprints. Thus the TPR is 90 percent and the FPR is 20 percent.

5.5.3 Detector Simulation

The focus of the evaluation was the localization module of NIDS. To compare the signalprint matching algorithm and data mining algorithm, we designed three detectors using LDA, QDA and signalprint matching algorithms respectively. To optimize the performance of the signalprint-matching-based detector, the matching rules used by the detector combined min-matches and max-matches.

Data Mining Method

Figure 5.6: *The evaluation of the data mining method*

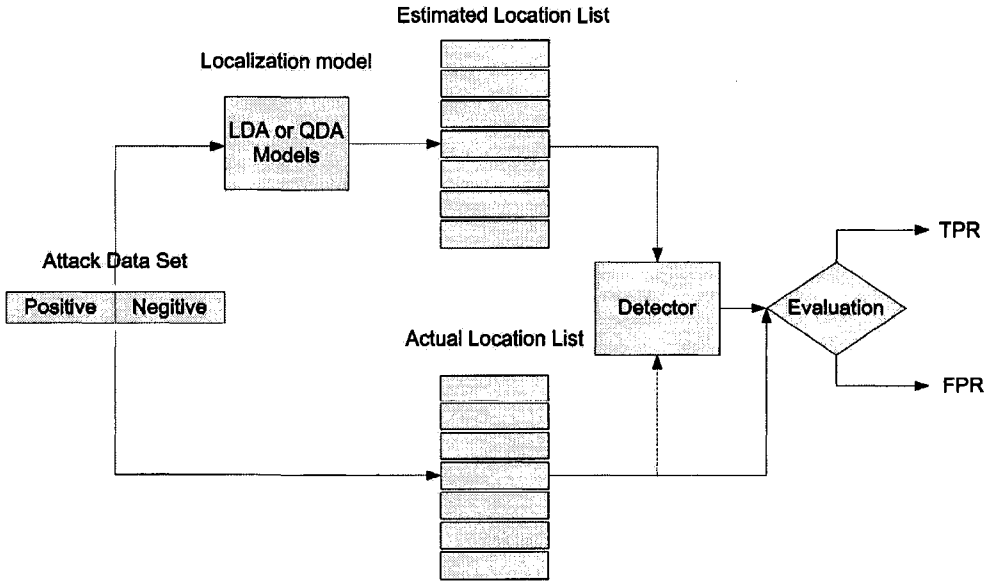


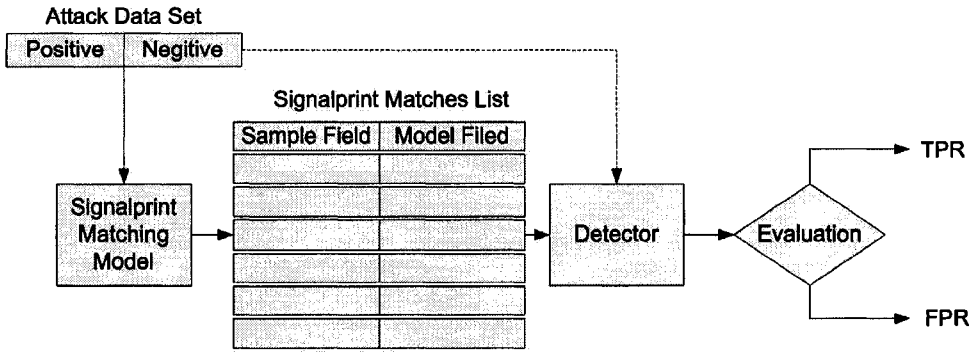
Figure 5.6 shows the simulation of a MAC address spoofing detector that was based on data mining localization model, and also shows the process of evaluation. The first step was training. The detector training included localization model training and attack detector training. The localization models were trained using the training data set as usual. The attack detector was trained for an individual attack data set separately. As mentioned previously, for fairly evaluating detectors, the evaluation was based on the average performance of multiple attack data sets. The data sets are generated for different attack locations. Each attack data set consisted of 50 percent signalprints representing attacks (positive) and 50 percent signalprints representing normal network traffic (negative). For simulation, instead of training, the detector used the actual location of each signalprint in each attack data set, and the positive locations and negative locations were clear.

After the detector training process, the multiple simulated attack data sets were applied to the detector one by one. The LDA- or QDA-based localization model estimated the location of every signalprint in the attack data set and generated an estimated location list. The list was sent to the attack detector and the detector reported all the suspicious signalprints to an evaluator. Meanwhile, the actual location list was sent to the evaluator. Then the evaluator computed the TPR and FPR according to the attack report and the actual location list of all signalprints.

Signalprint Matching Method

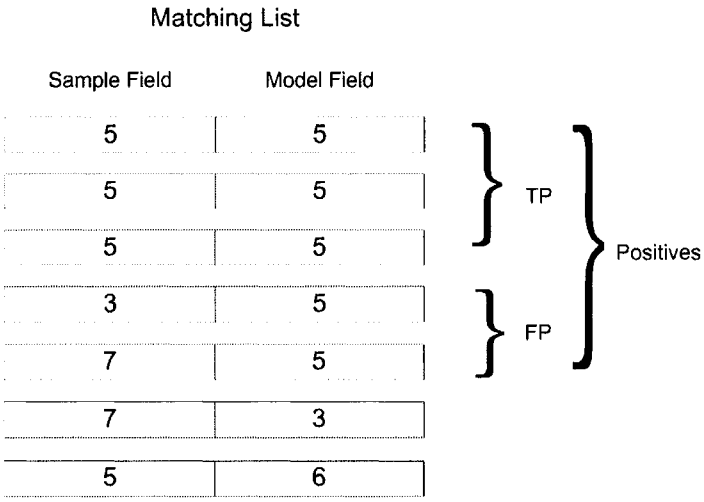
Figure 5.7 shows the simulation of a MAC address spoofing detector that is based on signalprints matching method, and the process of evaluation. The training process included signalprint-matching model training and attack detector training. The signalprint-matching model was trained by using the median value of training data set grouped by each location. The matching rules used by the signalprint-matching model were a combination of max-matches and min-matches in order to obtain the best matching result and lowest FP. The matching rules of the signalprint-matching model were adjustable through parameters, such as the number of max-matches and the number of min-matches that were required to find, and so on. The attack detector was trained using the same method as the attack detector in the simulation of data

Figure 5.7: *The evaluation of the signalprint-matching method*



mining based NIDS, which is using the attack and normal location information in each individual attack data set.

Figure 5.8: *Suppose the attack location has location code 5. The attack detector can easily detect suspicious matches. The evaluator computes the TPR and FPR based on the detection.*



Although the signalprint-matching model cannot estimate the location of a signalprint directly, the attack detector is able to detect suspicious signalprints by analyzing the matching list generated by the signalprint-matching model. Figure 5.8 shows how

the attack detector found suspicious matches using the matching list. The each entry of the matching list contained two fields used to identify two matched signalprints. The one that contains a signalprint in the attack data set is called sample filed; another one that contained one of the signalprints in the signalprint-matching model is called model filed, which is generated from the median value of the training dataset. Instead of using MAC addresses as identifiers of signalprints, we used the location code to identify signalprints in the simulation. The detector searched suspicious matches (positives) in the matching list of the model field that contains the location of attacks. If a suspicious match has the same location code in its sample field, then this indicates that we have a TP. Otherwise, the suspicious match is FP. The evaluator can compute the TPR and FPR based on the detection.

5.5.4 Confidence Interval

As discussed previously, the value of TPR and FPR is the average evaluation of multiple attack detections. For example, with the accuracy level of 2.4 meters (the threshold $a = 2.4$ meters), we evaluated 24 independent attacks. The values of TPR and FPR were various. We computed the confidence interval (CI) in order to evaluation the detection reliability of the detectors. Suppose μ is the mean, Let

$$\overline{X} = (X_1 + \cdots + X_n)/n,$$

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X},\right)^2.$$

Then

$$T = \frac{\overline{X} - \mu}{S/\sqrt{n}}$$

has a Students t-distribution whit n-1 degrees of freedom. If we want a CI with 95 percent confidence level, we have

$$Pr(-c < T < c) = 0.95$$

Consequently

$$Pr(\overline{X} - cS/\sqrt{n} < \mu < \overline{X} + cS/\sqrt{n}) = 0.95$$

The confidence interval is

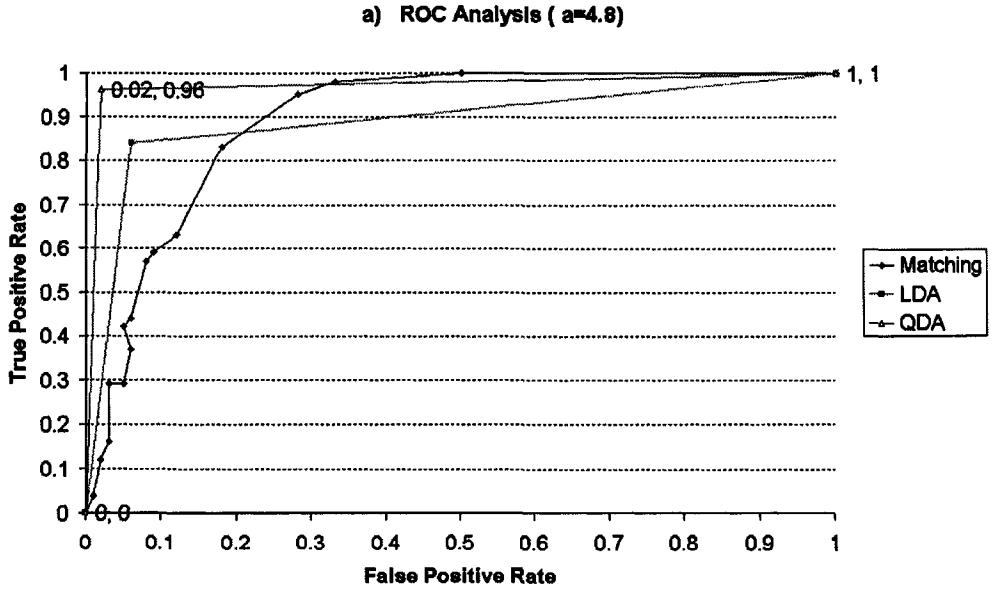
$$[\overline{X} - cS/\sqrt{n}; \overline{X} + cS/\sqrt{n}]$$

The c can be retrieved from *t distribution table* based on the confidence level and degrees of freedom.

5.5.5 The Comparison Results

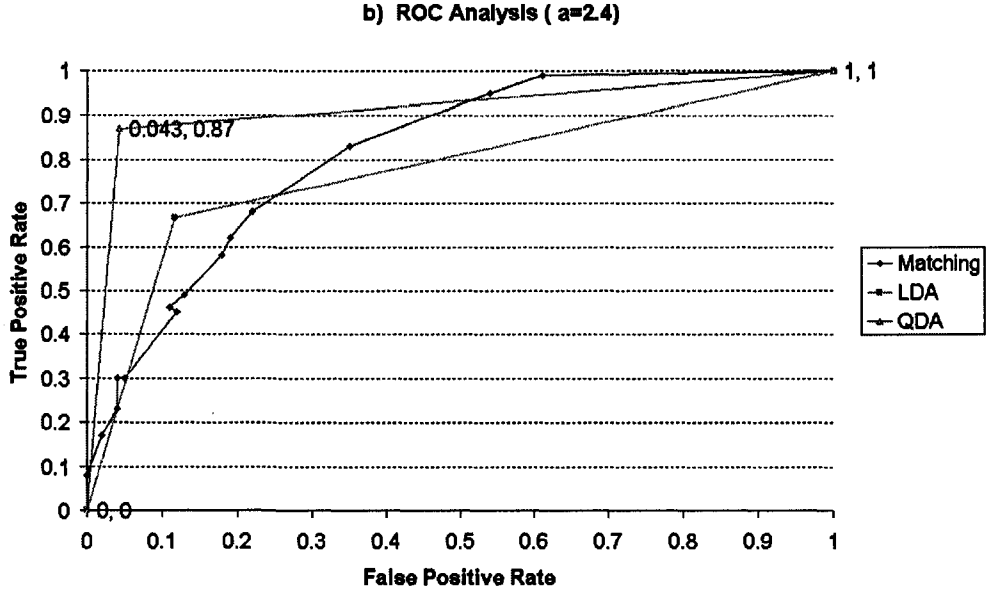
Signalprint Matching Method

Figure 5.9: *ROC analysis on LDA, QDA, and signalprint matching method scenario one. The QDA based detector has the best performance because it yields a point (0.02, 0.96), which represents 96 percent TPR and 2 percent FPR, in the ROC space. It is closest to the perfect point in the upper left corner (0, 1). The best operating point of signalprint matching method is (0.18, 0.83), which represents 83 percent TPR and 18 percent FPR. The matching rule at that point is* $maxMatch(S_1, S_2, 3) > 3 \wedge minMatch(S_1, S_2, 8) \geq 0$.



The ROC curves shown in Figure 5.9 illustrate the performance evaluation for MAC address spoofing detectors that are based on the LDA, QDA, and signalprint

Figure 5.10: *ROC analysis on LDA, QDA, and signalprint matching method scenario two. The performances are degraded as the separation distance of nodes decreases. The QDA based detector yields a point (0.043, 0.87), which represents 87 percent TPR and 4.3 percent FPR, in the ROC space. The best operating point of signalprint matching method is (0.35, 0.83), which represents 83 percent TPR and 35 percent FPR. The matching rule at that point is* $maxMatch(S_1, S_2, 3) > 3 \wedge minMatch(S_1, S_2, 8) \geq 0$.



matching method, respectively. The threshold for the generation of attack data set, $a = 4.8$ meters. Another ROC analysis result shown in Figure 5.10. The threshold, $a = 2.4$ meters.

Two detailed comparison tables are shown in Table 5.5 and Table 5.6. Table 5.5 shows the performance of the LDA and QDA-based detectors, and three signalprint-matching-based detectors using different parameters, when the threshold, $a = 4.8$ meters. The table lists the mean of TPR and FPR of multiple detection evaluations, and CI with 95 percent confidence level. Table 5.6 is based on the simulation with the threshold, $a = 2.4$ meters.

First of all, the data mining method is better than the signalprint-matching method. Not only because the data mining based detector is stable, but the TPR is much higher than the signalprint-matching-based detector when FPR is on the same

Table 5.5: *The performance of LDA, QDA, and signalprint matching detector. The a=4.8 meters, sample size is 10, and confidence level is 95 percent.*

	QDA	LDA	S-matching-1	S-matching-2	S-matching-3
TPR	0.958 ± 0.011	0.841 ± 0.042	0.828 ± 0.045	0.952 ± 0.024	0.635 ± 0.049
FPR	0.021 ± 0.007	0.063 ± 0.023	0.181 ± 0.046	0.28 ± 0.062	0.125 ± 0.034

Table 5.6: *The performance of LDA, QDA, and signalprint matching detector. The a=2.4 meters, sample size is 24, and confidence level is 95 percent.*

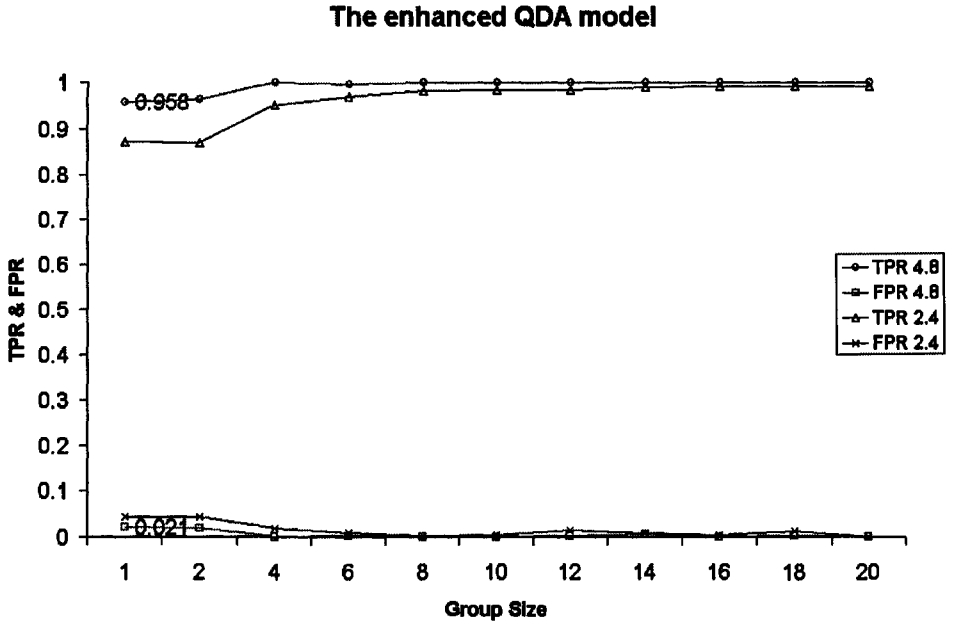
	QDA	LDA	S-matching-1	S-matching-2	S-matching-3
TPR	0.871 ± 0.009	0.668 ± 0.02	0.835 ± 0.013	0.953 ± 0.007	0.681 ± 0.015
FPR	0.043 ± 0.004	0.117 ± 0.008	0.353 ± 0.017	0.537 ± 0.02	0.225 ± 0.012

level. Unlike the signalprint matching detector, the QDA- and LDA-based detectors do not need to adjust several parameters to optimize the detectors in order to get the best performance. As long as the training data is good, the data mining model maintains a stable performance. The QDA method is the best among the three detectors. The result is consistency with our previous analysis for data mining model choice. Although the TPR of the signalprint-matching-based detector can be very high, it must trade off a very high FPR and it is unacceptable for a NIDS.

Another observation is that the threshold, a, is proportional to the performance of detectors. Because the threshold, a, is bigger in Figure 5.9, the performance of all detectors is better than the evaluation shown in Figure 5.10. For instance, the QDA-based detector has the best performance. The TPR is 0.96 and FPR is 0.02 in Figure 5.9, and the TPR is 0.87 and FPR 0.043 in Figure 5.10. Moreover, along with the increase in a, the performance of the QDA- and LDA-based detector are improving faster than the signalprint-matching-based detector.

The evaluation also shows that the TPR and FPR are not good enough for NIDS, at least when an attacker is physically very close to the victim nodes. The threshold, a, indicates the smallest distance between an attacker and targeted nodes when the evaluation is made. When the $a = 4.8$ meters, the QDA-based detector still has 0.02 FPR (false alarms) and the TPR is 0.96. Can the enhancement method achieve better performance?

Figure 5.11: *TPR grows rapidly as the group size increasing. Meanwhile, FPR becomes vary low. When the threshold, a=4.8 meters, the QDA-based detector achieves perfect performance with the group size of eight, TPR=1 and FPR=0. When the threshold, a=2.4 meters, the QDA-based detector achieves best performance with the group size of 16.*



5.5.6 The Enhancement Results

Since we have proved the enhancement method is able to achieve a high localization accuracy and low error rate, we applied the enhancement method to the QDA and LDA detectors and evaluated their performance using the attack simulation. Table 5.7 *a*) lists the TPR and FPR values for a group size of one to 20 with the threshold, $a = 4.8$ meters. The IC was calculated based on a 95 percent confidence level, and the number of simulation was 10. The Table 5.7 *b*) shows the TPR and FPR values for a group size of one to 20 with the threshold, $a = 2.4$ meters. The IC was calculated based on a 95 percent confidence level, and the number of simulation was 24.

The evaluation can prove that the enhancement method has a better performance than the normal QDA and LDA detectors. The enhancement method based on a QDA algorithm has the best TPR and FPR. Figure 5.11 shows the TPR and FPR of the

Table 5.7: *Evaluation of the enhancement method of attack simulation*

a) The enhancement method evaluation (a=4.8m, sample size=10, Confidence Level=95%)

GroupSize	2	4	6	8	10	12	14	16
LDA								
TPR	0.861±0.034	0.934±0.022	0.949±0.02	0.968±0.022	0.966±0.018	0.97±0.016	0.965±0.021	0.975±0.016
FPR	0.049±0.013	0.025±0.008	0.021±0.009	0.01±0.006	0.016±0.008	0.031±0.01	0.015±0.005	0.005±0.004
QDA								
TPR	0.963±0.01	0.999±0.001	0.995±0.001	1±0	1±0	1±0	1±0	1±0
FPR	0.019±0.006	0±0	0.003±0.001	0±0	0±0	0.013±0	0.006±0	0±0

b) The enhancement method evaluation (a=2.4m, sample size=24, Confidence Level=95%)

GroupSize	6	8	10	12	14	16	18	20
LDA								
TPR	0.808±0.023	0.835±0.024	0.831±0.025	0.845±0.026	0.846±0.025	0.838±0.025	0.841±0.026	0.839±0.026
FPR	0.072±0.009	0.058±0.009	0.066±0.01	0.067±0.01	0.074±0.01	0.06±0.01	0.071±0.01	0.063±0.01
QDA								
TPR	0.97±0.005	0.981±0.004	0.983±0.004	0.984±0.005	0.989±0.003	0.992±0.002	0.991±0.003	0.992±0.003
FPR	0.009±0.001	0.003±0.001	0.004±0.001	0.014±0.001	0.01±0.001	0.004±0.001	0.012±0	0±0

QDA-based detector improve rapidly as the group size increasing. The detailed values of TPR and FPR are listed in Table 5.7. In Table 5.7 *a*), for a QDA-based detector, the TPR achieved 1 when the group size was eight and the FPR was zero. However, it is not true that a group size lager than eight can always get such high performance. It was different from our previous evaluation when the enhanced method was applied to the localization model.

We observed that when the group size was too big, the FPR oscillated around zero but did not stay on zero. This was caused by the feature of the enhancement method. When we evaluated the performance of the enhancement method applied in a NIDS, unlike the data set for the evaluation of a localization model, an attack data set was used. In the attack data set, the signalprints related to attacks and the normal network traffic were mixed up. When the detector grouped the signalprints based on the MAC address, in each group, the ratio of signalprints related to attacks and signalprints related to the victim node changed from time to time because of the

forged MAC address. So when the detector grouped the signalprints, a group that was too large will have a risk of degrading the performance of the detector, because the detecting once gets a false result for one group, the bigger size of the group causes a higher FPR.

The evaluation shows that the QDA-based detector achieves best performance when the group size is 8 and 16 depending on the different accuracy levels. When the separation distance of any neighboring nodes is 2.4 meters, the TPR of the detector is $[0.992 \pm 0.003]$ and the FPR is $[0.004 \pm 0.001]$ by using 16 signalprints. When the separation distance of any neighboring nodes is 4.8 meters, the TPR of the detector is $[1 \pm 0]$ and FPR is $[0 \pm 0]$ by using 8 signalprints. The CI is calculated based on a 95 percent confidence level.

Chapter 6

Conclusions and Future Initiatives

The conclusions are presented in this chapter. With suitable methodologies, an RSS-based NIDS can be used to detect attacks exploiting the vulnerabilities of the low network layer of WiFi/802.11 networks. Although the detecting performance is constrained by a fundamental limitation because of the nature of radio waves, enhancements to improve the performance of NIDS are feasible. Meanwhile, further research is required in order to: test the efficiency of the detecting model in real-time systems, analyze the influences of the network environment change, and optimize the model training process.

6.1 The Accuracy Limitation

A problem of the RSS-based NIDS and indoor localization model is the high error rate. According to the results of our experiments, if we assume that all the network nodes do not change their antenna orientations, the localization model can correctly distinguish (with zero error rate) the network nodes with a separation distance of 4.8 meters. But with a decrease of the separation distance, the error rate keeps rising. The error rate is 2.4 percent at 2.4 meters separation distance, and 7.4 percent at 1.2 meters separation distance.

The results prove the accuracy limitation of RSS-based localization algorithms for indoor WiFi networks. The limitation introduced by Elnahrawy et al. [9] shows that with an acceptable error, the distinguishable separation distance between network nodes is three meters. Similar limitations can be found in other researches. [10][2][30]. Moreover, if we consider that the antenna orientation of wireless nodes may change from time to time, the results are much worse. The results of our experiments showed that: the best localization model, which is based on QDA algorithm, still has a 3.8 percent error rate when the separation distance is 4.8 meters; if the separation distance

decreased to 1.2 meters, the error rate of the best localization mode is 33.4 percent.

The high error rate of the localization model is unacceptable for an NIDS. The high false alarm rate must be avoided for an anomaly-based NIDS. We need a low error rate, even zero error rate localization method.

6.2 Enhancement Result

An enhancement of localization was proposed and tested. The enhanced localization model can give a zero error rate. The enhancement method is based on the observations of the *confusion matrix* built on the estimated localization and the actual location of signalprints. Although the error rate if pretty high, the misclassified location results are clustering around the actual location of signalprints. Instead of using individual signalprint, we use a small group of signalprints that are supposed related to a same node as input date and find the most probable location as the result to enhance the accuracy of localization model. There is a trade off between accuracy and efficiency.

The enhancement localization model gives a zero error rate even if we allow the network nodes to change their antenna orientations freely. Under a reasonable separation distance (i.e., 2.4 meters), the localization model achieves a zero error rate when the size of the group is 18. If we have a separation distance 1.2 meters, the model requires a group of signalprints of size 44 to achieve a zero error rate.

Indeed, the enhancement method also creates a problem when it is applied to a NIDS. Since the enhancement uses a group of signalprints instead of the signalprints individually, the signalprints representing the packets sent by an attacker could be ignored by the localization model if an attacker launches the attack by only sending a small amount of packets or keeps sending malicious packets at a very low frequency. We can mitigate this bad side-effect of the enhancement method by using another observation on the *confusion matrix*. The normally mislocalized signalprints do not fall into locations far from the actual correct locations. The range is about ten meters. This range actually is the RSS oscillation range shown in the Section 5.4. If we assume that the attackers and the targeted nodes have a bigger distance than the localization error, ten meters, then we can detect those attacks by capturing the

signalprints related to the location that have an abnormally large distance to others in a same signalprint group. Those abnormal signalprints and their location are sent to the attack detectors of NIDS and are investigated by the detectors.

6.3 Advantage of RSS-based NIDS

A RSS-based NIDS detects suspicious network activities based on the physical location of network nodes. That makes the NIDS sensitive for all identity-based attacks in WiFi/802.11 and WiFi/802.11i networks. Location information of a node is a simple feature. The anomaly pattern is able to build upon the location information very quickly.

We believe a signalprint is hard to manipulate by attackers. A signalprint is built on the RSS values measured by several monitors. The factors affecting the value of RSS are unique for each monitor. Unless an attacker takes control of all monitors, the signalprints are immune to being forged.

Since we have the localization model, we not only can localize the authorized clients, but can pinpoint the location where an attack is launched. This capability is useful to design a disaster recovery mechanism. The location of the problem usually can be detected by analyzing the statistical logs about the suspicious packets and their sender's location, even the network communication is blocked by serious DoS attacks.

6.4 Detected Attacks

Based on the enhanced localization model, the NIDS focused on MAC address spoofing attack. Basically, a MAC address spoofing attack is suspected if more than one network device with identical MAC address appear to be at different locations, or the NIDS detects that several packets with different MAC addresses appear to be sent from the same location. Considering that the location of a network device may change, the NIDS uses a list of possible locations of each node instead of a single one.

For the other low layer attacks, fortunately, the analysis in Section 2.2 shows that the detection of a MAC address spoofing attack is the cure for most of the

MAC layer attacks in a WiFi/802.11 or WiFi/802.11i networks, such as rogue AP, jamming, session hijacking, and various DoS attacks. Moreover, because of protection from session hijacking or de-authentication and de-association attacks, the NIDS is able to defeat some tools that are used to accelerate the WEP or WAP key breaking process.

6.5 Future Initiatives

Future experiments are needed for testing the efficiency and effectiveness of the NIDS model in real-time systems. On one hand, the theoretical running time of detection is $O(gm^2n) + O(n)$. The m is the number of monitors, the g is the size of the signalprint group, and the n is the number of locations that are available for network nodes. Since the m and g are usually constant numbers, the running time can be $O(n)$, and it depends on the area of the monitored network. On the other hand, the waiting time for grouping signalprints will delay the detecting process. The trade off between accuracy and efficiency needs to be further tested.

A feasible model training procedure needs to be designed. The model training includes localization model training and intrusion detector training. Once there is a big change in a network environment or new wireless nodes are added, the localization model and detector need to redo the training or at least make an adjustment. Since repeated training is inevitable, the time of training is required to be as short as possible. The optimized value of training data size needs to be tested.

Moreover, the change of network environment is one of the challenges for an indoor RSS-based localization. Our model covers the changes of antenna orientation and the changes of physical location. But there are some other factors that may influence the accuracy of a localization model, such as the number of APs, the basic noise level, and even the movement of people, furniture, and so on.

Appendix A

Glossary

AP	Access Point
CI	Confidence Interval
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access With Collision Avoidance
CTS	Clear to Send
DCF	Distributed Coordination Function
DHCP	Dynamic Host Conifguration Protocol
DoS	Denial of Service
ESS	Expanded Service Set
FP	False Positive
FPR	False Positive Rate
IDS	Intrusion Detection System
IT	Information Technology
IV	Initialization Vector
LDA	Linear Discriminant Analysis
MAC	Media Access Control
MIC	Message Integrity Check
NAV	Network Allocation Vector
NIDS	Network Intrusion Detection System
PDA	Personal Digital Assistant
PMK	Pairwise Master Key
PSK	Pre-shared Key
PTK	Pairwise Transient Key
QDA	Quadratic Discriminant Analysis
RC4	Rivest Cipher
RFF	RF Fingerprint
ROC	Receiver Operation Characteristic
RSN	Robust Security Network
RSS	Received Signal Strength

RTS	Request to Send
SIFS	Short Inter-frame Space
TOA	Time of Arrival
TP	True Positive
TPR	True Positive Rate
WEP	Wired Equivalent Privancy
WLAN	Wireless Local Area Network
WPA	WiFi Protected Access
XOR	Exclusive OR

Bibliography

[1] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. Enhancing the security of corporate WiFi networks using DAIR. In *MobiSys '06: Proceedings of the 4th international conference on mobile systems, applications and services*, pages 1–14, New York, NY, USA, 2006. ACM Press.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[3] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing user behavior and network performance in a public wireless LAN. In *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on measurement and modeling of computer systems*, pages 195–205, New York, NY, USA, 2002. ACM Press.

[4] M. Barbeau. WiMAX/802.16 threat analysis. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on quality of service & security in wireless and mobile networks*, pages 8–15, New York, NY, USA, 2005. ACM Press.

[5] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the twelfth USENIX security symposium*, pages 15–28, Washington, DC, USA, Aug. 2003. USENIX Association.

[6] H. Berghel and J. Uecker. WiFi attack vectors. *Commun. ACM*, 48(8):21–28, 2005.

[7] Bhagyavati, W. C. Summers, and A. DeJoie. Wireless security techniques: an overview. In *InfoSecCD '04: Proceedings of the 1st annual conference on information security curriculum development*, pages 82–87, New York, NY, USA, 2004. ACM Press.

[8] Cisco. Addressing wireless threats with integrated wireless IDS and IPS in the Cisco unified wireless network. `http://www.cisco.com/application/pdf/en/us/guest/products/ps6521/c1244/cdccont_0900aecd804f155b.pdf`, 2007.

[9] E. Elnahraway, X. Li, and R. P. Martin. The limits of localization using RSS. In *SenSys '04: Proceedings of the 2nd international conference on embedded networked sensor systems*, pages 283–284, New York, NY, USA, 2004. ACM.

[10] D. B. Faria and D. R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security*, pages 43–52, New York, NY, USA, 2006. ACM Press.

[11] D. B. Faria and D. R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *WiSe '06: Proceedings of the 5th ACM workshop on wireless security*, pages 43–52, New York, NY, USA, 2006. ACM.

[12] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:57–90, 1936.

[13] F. Guo and T. cker Chiueh. Sequence number-based MAC address spoof detection. In *8th International Symposium on Recent Advances in Intrusion Detection (RAID 2005)*, 2005.

[14] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on mobile computing and networking*, pages 70–84, New York, NY, USA, 2004. ACM Press.

[15] J. Hall, M. Barbeau, and E. Kranakis. Detecting rogue devices in bluetooth networks using radio frequency fingerprinting. In *Communications and Computer Networks*, pages 108–113, 2006.

[16] R. Hytnen and M. Garcia. An analysis of wireless security. *J. Comput. Small Coll.*, 21(4):210–216, 2006.

[17] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In *E-WIND '05: Proceeding of the 2005 ACM SIGCOMM workshop on experimental approaches to wireless network design and analysis*, pages 11–16, New York, NY, USA, 2005. ACM Press.

[18] T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless LANs. In *MobiSys '07: Proceedings of the 5th international conference on mobile systems, applications and services*, pages 246–257, New York, NY, USA, 2007. ACM Press.

[19] P. Krishnan, A. S. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu. A system for lease: Location estimation assisted by stationary emitters for indoor RF wireless networks. In *INFOCOM*, 2004.

[20] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, D. S. Wallach, and G. Marceau. Robotics-based location sensing using wireless ethernet. In *MobiCom '02: Proceedings of the 8th annual international conference on mobile computing and networking*, pages 227–238, New York, NY, USA, 2002. ACM Press.

[21] G. Lehembre. WiFi security - WEP, WPA and WPA2. *Hakin9*, 2006.

[22] MadWiFi.org. http://madwifi.org/.

[23] S. Patton, W. Yurcik, and D. Doss. An achilles' heel in signature-based IDS: Squealing false positives in snort. http://citeseer.ist.psu.edu/patton01achilles.html, 2001.

[24] B. Potter. Wireless hotspots: petri dish of wireless security. *Commun. ACM*, 49(6):50–56, 2006.

[25] r project.org. http://www.r-project.org/.

[26] S. Radosavac, J. S. Baras, and G. V. Moustakides. Impact of optimal MAC layer attacks on the network layer. In *SASN '06: Proceedings of the fourth ACM workshop on security of ad hoc and sensor networks*, pages 135–146, New York, NY, USA, 2006. ACM Press.

[27] RFC4017. http://www.apps.ietf.org/rfc/rfc4017.html.

[28] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker. Mojo: a distributed physical layer anomaly detection system for 802.11 WLANs. In *MobiSys '06: Proceedings of the 4th international conference on mobile systems, applications and services*, pages 191–204, New York, NY, USA, 2006. ACM Press.

[29] M. Siyal and F. Ahmed. *HandBook of Wireless Local Area Networks*, chapter 14, Security Services and Issues in WLANs. Taylor & Francis Ltd., 2005.

[30] P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach. Wireless LAN location-sensing for security applications. In *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, pages 11–20, New York, NY, USA, 2003. ACM Press.

[31] WepLabProject. http://weplab.sourceforge.net/.

[32] wirelessdefence.org. http://www.wirelessdefence.org/Contents/AircrackMain.htm.

[33] wireshark.org. http://www.wireshark.org/.

[34] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on mobile ad hoc networking and computing*, pages 46–57, New York, NY, USA, 2005. ACM Press.

[35] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless LAN monitoring and its applications. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 70–79, New York, NY, USA, 2004. ACM Press.

[36] M. Youssef and A. Agrawala. The horus WLAN location determination system. In *MobiSys '05: Proceedings of the 3rd international conference on mobile systems, applications, and services*, pages 205–218, New York, NY, USA, 2005. ACM Press.

[37] Y. Zahur and T. A. Yang. Wireless LAN security and laboratory designs. *J. Comput. Small Coll.*, 19(3):44–60, 2004.

[38] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac. Indoor location tracking using RSSI readings from a single WiFi access point. *Wirel. Netw.*, 13(2):221–235, 2007.