MITIGATION OF PRIMARY USER EMULATION ATTACKS USING BELIEF PROPAGATION

Sasa Maric

Masters in Research and Philosophy (MRES)

Department of Electronic Engineering
Macquarie University

October 10, 2014

Supervisor: Professor Sam Reisenfeld

ACKNOWLEDGMENTS

I would like to express my appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my supervisor Professor Sam Reisenfeld, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project.

STATEMENT OF CANDIDATE

I, Sasa Maric, declare that this report, submitted as part of the requirement for the award of Masters in Research and Philosophy, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment an any academic institution.

Student's Name:

Student's Signature:

Date:

ABSTRACT

Cognitive Radio (CR) is a promising technology that has the potential to revolutionize spectrum utilization by enabling unlicensed users (secondary users) to seek opportunities for transmission by taking advantage of the idle periods of licensed users (primary users) [1]. However, participants in the CR network can comprise of malicious adversaries who adopt dishonest and non-cooperative strategies to attack the network. An attack called the Primary user emulation attack has been identified as the most serious threat to cognitive radio security. In primary user emulation attacks a malicious user emulates the characteristics of a primary user and transmits on available frequency spectrum channels. As a result, secondary users are tricked into believing that the channel is occupied and avoid it, which allows the malicious user to occupy the entire frequency spectrum band uncontested. This report proposes a new technique based on belief propagation to combat primary user emulation attacks in cognitive radio networks. We introduce a method that dramatically reduces the computational complexity and run time of the existing techniques, while also improving the performance.

Contents

List of Figures

Chapter 1

Introduction

Cognitive Radio (CR) is a promising technology that has the potential to alleviate the spectrum shortage problem that has been caused by our ever increasing demand for communication services. Due to our growing reliability on wireless devices, we have reached a point where frequency has become scarce. Currently, most of the frequency spectrum has been assigned to existing communication services resulting in very little room for future expansion of new wireless communication links and networks.

Traditional spectrum allocation methods allocate spectrum over large regions and time spans to primary users, which are licensed by a government regulatory office, such as the Federal Communications Commission in the United States. Channels in the licensed spectrum bands are allocated exclusively to primary users and are inaccessible to other users [2]. Users, other than primary users who could potentially use these channels, are called secondary users. It has been showed that this traditional allocation method of fixed channel allocation to primary users is leading to a very low utilization across the licensed spectrum [3] [4]. Cognitive Radio, a collection of intelligent methods designed to use the radio spectrum in an efficient and dynamic manner, has been proposed as a solution to the frequency spectrum shortage. Cognitive Radio proposes to increase the efficiency of radio spectrum use by allowing secondary users to use channels when they are unoccupied by primary users. In this way, the average percentage of time for which the channels are actively carrying communication signals is increased. As a result, the total data throughput for the same bandwidth allocation is also increased. Ideally, this increased efficiency should be obtained without the secondary users causing interference to the communications of primary users [2].

Despite its tremendous potential, Cognitive Radio is yet to be accepted as the solution to the radio spectrum shortage problem that exists today. One of the reasons for this is cognitive radio networks are susceptible to a number of jamming attacks. One of the most exploited areas in cognitive radio is the spectrum sensing phase, where secondary users scan the frequency spectrum looking for available channels which are unoccupied by primary users. During this phase if an attacker was able to mimic the signal prop-

erties of a primary user, they would be able to trick the secondary users into believing that available channels are being used by primary users, this would insure that secondary users vacate the channels leaving them available for malicious users to utilize uncontested. This form of attack is attack is called a Primary User Emulation Attack (PUEA).

It was initially thought that cognitive radios may enjoy better anti-jamming capability then conventional networks because of their flexible physical and MAC layer functions[5]. However this has been proven to be false, in fact it has been show that cognitive radios will have to deal with a larger variety of attacks then conventional networks [5]. One of the reason for this, is that secondary users can only use radio frequency bands that are vacant. This provides jammers a low cost and easy way disrupt the system, by jamming multiple frequency bands the can cause serious performance degradation[5][6].It is then essential, that security mitigation techniques are implemented early in the development of cognitive radio to combat these threats. There are a number of potential threats that CRs will have to face. These include attacks such as the Spectrum Sensing Data Falsification Attacks, where attackers try and pass false information around the network to cause interference to primary users and reduced bandwidth capacity for secondary users. Other attacks like the hello attack and the sink hole attack target particular OSI layers to try and disrupt communication between secondary users. Another popular attack is the lion attack which is a combination of attacks designed to have multiple layer impact. These are further discussed in chapter 3. It has been shown that attacks such as the primary user emulation attacks can have severe effects on the overall performance of the network [5][6][7]. It is essential that reliable and accurate methods for the extenuation of these attacks are found.

This paper proposes a new technique that reduces the effects of primary user emulation attacks, based on a belief propagation technique proposed in [2]. A popular way to overcome jamming attacks is to use Received Signal Strength (RSS) measurements at cooperating secondary users [2][8] to localise transmitters. The author in [2] presents a mitigation scheme where each SU receives a RSS signal from a transmitter. Using these measurements each SU determines an approximate location of the transmitter. Since the locations of primary users are assumed to be known by secondary users, transmitter locations that do not corresponding to these primary user locations are identified as jammers. Each secondary user calculates an approximate belief about the probability of whether a suspect is a primary user or an attacker using an algorithm called belief propagation. Each secondary user will forward their local beliefs to all their neighbours in the form of a message. After all the messages have been exchanged, each secondary user computes their final belief using their own local belief and the product of all the beliefs from all its neighbors [2].

The belief propagation technique proposed in [2] presents an effective and reliable method for identifying malicious users in a CR network. However, the current algorithm has a two major deficiencies. The first is its high complexity and slow convergence time, this is due to an overcomplicated local function. As a result, as the number of secondary

users in the network increases, the convergence time of the algorithm increases exponentially.The second deficiency is the result of a poorly defined compatibility function which reduces the level of cooperation between secondary users and causes a reduction in performance. This paper proposes a new technique based on [2] that uses a simplified algorithm to calculate the local function at each user. The modified algorithm insures that the computational complexity and run time increase linearly instead of exponentially. As a result, we are able to significantly reduces the amount of time the new algorithm takes to converge. In addition to the improved efficiency, the new algorithm introduces a modified compatibility function that increases the level of cooperation between secondary users. This results in an improvement in the accuracy and reliability of the algorithm, which enables for better detection of malicious users. We believe that the findings of this paper present a significant step forward in the mitigation of primary user emulation attacks using belief propagation.

1.1 Project Goals/ Motivation

Successful deployment of cognitive radio technology and the realization of its benefits will depend on the placement of essential security mechanisms. It is important that security threats to cognitive radios are dealt with before cognitive radio networks are deployed. This project aims to develop a new technique to mitigate against primary user emulation attacks(PUEA). The project aims to study existing techniques and methods and develop an improved method to combat this type of attack.

The performance of the new technique will be investigated through MATLAB simulations. The new technique will be evaluated according to two key attributes, computational complexity and accuracy. We aim to identify the weaknesses and strengths of existing methods and use this information to develop a new, more efficient algorithm to combat PUEA. It is important that the new technique is able to improve the performance of existing techniques in one of two ways:

- Improve the computational efficiency of an existing algorithm allowing for simpler and faster detection of malicious users.
- Improve the accuracy of an existing technique insuring reliable detection of malicious users.

We assume that for a technique to be acceptable it must not only improve one attribute of an existing technique but also achieve satisfactory results in the other. If for example, the new technique improves the computational complexity, it must also achieve satisfactory levels of accuracy. This project aims to introduce a new technique that is capable of detecting malicious users in the most efficient and accurate way possible.

1.2 Background and Related work

In recent years there has been an increased amount of research into cognitive radio networks. However, there has been very little research into cognitive radio security and the effects that malicious users have on cognitive radio networks. Primary user emulation attacks have been identified as a serious threat to cognitive radio (CR) security. In a primary user emulation attack a malicious user emulates the signal characteristics of a primary user. The goal of the malicious user is to trick secondary users into thinking that he is a primary user and that he is active on the band. If he is successful, the secondary users will vacate the band and the malicious user will have the entire frequency band to use uncontested [9][10][11][2]. A number of researchers have shown that primary user emulation attacks severely decrease the performance of the network and because of this it has been identified as the most serious type of attack against cognitive radio networks[12][13][2].

A number of mitigation techniques have been proposed to combat primary user emulation attacks. The most promising of these use localization of the transmitter. There are two prominent methods for characterising transmitters using RSS(Received Signal Strength) measurements the first approach uses secondary user cooperation, this type of method is classified as the distributed method and involves secondary users trying to solve the localization problem individually using information from cooperating nodes. The other approach is the central method, in this method nodes are scattered around the network and collect snapshots of the transmitted signal. These measurements are sent to a central node that processes the information and makes a decision on whether the suspect is a legitimate user or an attacker.

In [2] a technique based on secondary user cooperation is presented. This technique relies on a triangulation approach to localize an incoming signal. The triangulation technique is based on interpretation of the received signal power (RSS) of the incoming signal. Each secondary user uses the RSS value of the incoming signal to compute the distance of the attacker. Since the secondary user does not know which direction the transmitter is sending from they are not able to locate the transmitter. However, they are able to formulate a circle that has a radius equal to distance that the secondary user thinks that the transmitter is transmitting from. By itself this result does not provide a location for the transmitter. However, if three cooperating secondary users all formulate individual circles according to their RSS measurements, the three circles should intersect at a single point which would give the location of the transmitter. One of the assumptions of this algorithm is that the location of the primary user is known. To identify a malicious user, secondary users compare the location of the transmitter to the location of the primary user. If the two are the same the suspect is assumed to be a valid primary user, if not they are assumed to be a malicious attacker.

The main deficiency of this method is that it does not work well when shadowing

is introduced. When shadowing is introduced the circles that are produced by the secondary users do not overlap at a single spot making it impossible to determine the location of the transmitting signal. Another disadvantage is that this technique relies on cooperative measurements from at least four secondary users. If a secondary user does not have three other SU close to it, the localization of the transmitter cannot be done. The triangulation technique is very simple, very efficient and very accurate if no shadowing is present. However, because of its susceptibility to shadowing it is not suitable for practical implementation.

Locdef[10] is a localisation method that uses both localization of the transmitter and signal characteristics to determine if the transmitter is a malicious user or not. The Locdef scheme uses sensor nodes scattered around the network to take snapshots of the incoming Received Signal Strength (RSS) at different locations in the network. These measurements are sent to a central location for processing. By identifying peaks in the RSS, a central node is able to determine the location of the transmitted signal. Locdef uses a three stage verification scheme to determine the validity of the incoming signal. The first stage of the Locdef scheme looks at the RSS of the signal to determine if it is coming from a primary user location or not. In the second stage the receiver looks at the energy of the received signal. The reason for this is that secondary users are not able to transmit at high power levels, whereas primary users often are. If a suspect passes the first two stages, the scheme moves on to the last stage where it compares the signal characteristics of the incoming signal with the known characteristics of the idle primary user. If the characteristics of the incoming signal do not match the known signal characteristics of the primary user, the transmitter is deemed to be a malicious user.

The Locdef scheme is very reliable and accurate. This is due to its multi-stage verification scheme which insures that the results obtained by the technique are very accurate. However, the scheme itself is very complicated. For reliable and accurate results the scheme relies on a large number of sensors that collect a large amount of data, this data must be processed in a timely manner which means that processing equipment has to be fast. This makes its implementation both complicated and expensive to implement.

In [14] a scheme based on a combination of two signal characteristic comparison methods is presented. This technique combines two methods called time difference of arrival (TDOA) and frequency difference of arrival (FDOA) to determine the location of the incoming signal. TDOA uses the differences in the time delay of signals arriving at secondary user stations to determine the location of a transmitter. TDOA uses four receiving stations that use three dimensional time differences of four stations to get the positioning equations [14]. FDOA is used to estimates the location of target by the Doppler frequency [14]. Individually neither technique is capable of reliably locating the transmitter. However, when used together, TDOA provides basic positioning points that are used by FDOA to determine the exact location of the transmitter. This technique is very accurate and works well with both stationary and moving targets. However, it requires complex

equipment at the receiving station. Its high level of complexity means that it is expensive and complicated to implement and run.

Papers [13] and [15] present two primary user emulation attack mitigation schemes based on authentication and encryption. In [15] the author outlines a centralised scheme in which each primary user is given a unique ID number and a random variable (HM) by a centralized base station. Every time a suspect becomes active, the base station goes through a two-step authentication process to insure that the suspect is a valid primary user. Before a primary user can access the network, the user must send their ID number to the BS for authentication. The primary user ID is compared to a pool of identification numbers that correspond to all primary users in the area. If the ID number corresponds to one of the ID numbers in the pool, the scheme moves on to step two of the authentication process. If it does not, the user is treated as a malicious user and is ignored. The second step of the process is called the information displacement step. In this step the HM variable is multiplied by an encryption matrix which returns a value M that is compared to a set of expected values. If the value corresponds to the expected values, the transmitter is authenticated as a primary user. If it does not, the transmitter is treated as a malicious user and is ignored.

A major issue with this technique is that it requires heavy participation from the primary user. According to the Federal Communications Commission (FCC) a CR user should operate on a non-interference basis with primary users[16]. A primary user should not have to do any extra work or authenticate itself to secondary users. Primary user operation should be kept as uninterrupted as possible. This technique clearly violates that requirement making it impractical for implementation.

In [2] the author presents a technique based on belief propagation. This technique uses cooperation between secondary users to localise a transmitter. Comparing this to the known location of a primary user each secondary user is able to determine with a certain probability whether the transmitter is a primary user. The author denotes this probability as a belief. Secondary users in the network calculate their own local belief and exchange them to their neighbours. Then, each secondary user calculates a final belief using its own beliefs and all the beliefs from its neighbours. This paper modifies the algorithm described in [2] and suggests a useful procedure for determining whether the received signal originates from an attacker or not. Our paper presents substantial improvements to the algorithm described in [2].

1.3 Contribution

The foremost argument against the implementation of cognitive radio is its poor security framework and the lack of efficient techniques to deal with the variety of security threats imposed on cognitive radio networks. This project aims to contribute to the

development of the security framework of cognitive radio networks by introducing a new technique that improves detection of malicious users in cognitive radio networks. We hope to create a technique that will accurately and efficiently moderate the effects of primary user emulation attacks on cognitive radio networks. In doing so, this project hopes to contribute to the improved performance of cognitive radio networks.

Chapter 2

Cognitive Radio Technology

A cognitive radio is an intelligent radio that can be configured dynamically. It increases spectrum efficiency by allowing unlicensed users to use parts of the spectrum when licensed users are idle. In the cognitive radio literature a licensed user is referred to as a primary user (PU) and an unlicensed user is referred to as a secondary user (SU). A SU scans parts of the spectrum looking for bands where a PU is idle. When an idle band is found the SU is able to use the band as long as the licensed user is idle. It is essential that the SU is constantly monitoring the state of the licensed user. If a licensed user becomes active the channel must be vacated immediately. The International Telecommunications union (ITU) defines a cognitive radio as a radio system employing technology that allows the system to obtain knowledge of its operational and geographical environment, established policies and its internal state, to dynamically and autonomously adjust its operational parameters and protocols according to its obtained knowledge in order to achieve predefined objectives, and to learn from the results obtained [17].

This chapter provides an overview of cognitive radio. It presents an introduction into the basic concepts and theories associated with the function and operation of cognitive radio networks. The primary objective of cognitive radio is to provide a means to utilize the radio frequency spectrum more efficiently. This is achieved by allowing secondary user access to frequency bands that were originally designated for primary users. This insures that frequency bands that were previously underutilized for large amounts of time are now utilized by secondary users. Secondary users must continually monitor their surrounding environment to insure that a primary user has not become active. In order to achieve this, a secondary user must be able to recognize primary users on the network. It is important that a secondary user is able to distinguish between a primary user and other secondary users. A secondary user must also be aware of potential malicious nodes that try to mimic the signal properties of primary users in order to gain access to a frequency band uncontested.

To successfully achieve its primary goals it is essential that cognitive radio users are able to cooperate with each other. Cooperation is defined as a paradigm that allows

distributed terminals in a wireless network to communicate through some distributed transmission or signal processing so as to realize a new form of space diversity to combat the detrimental effects of fading channels [18]. Cooperation between nodes has been shown to increase the spectrum access efficiency and improve network performance [19]. In a cooperative network each secondary node makes independent observations about the network environment. These observations are then distributed around the network in a broadcast manner so that all SUs in the network are aware of changing conditions. Cooperation is critical in insuring that secondary users are able to provide high quality accurate and effective services.

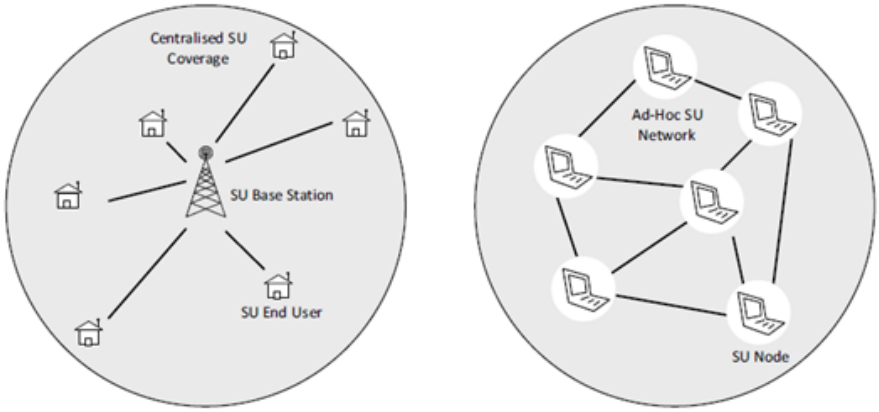
This chapter is organised into five sections that provide a basic CR framework. Part one presents an overview of cognitive radio architecture, part two introduces the cognition cycle, part three presents an overview of the various spectrum sensing techniques, part four provides insight into the role of SDR in cognitive radio networks and part five talks about cooperation in CR networks.

2.1 Cognitive radio Architecture

Cognitive Radio networks do not have a fixed architecture. Rather, CR architecture depends on the application that CRs are used for. This allows cognitive radio networks to be extremely flexible and provide high functionality and efficiency to its users. The two most predominant architectural models used by cognitive radio networks are the centralised architecture and the distributed architecture [20]. The centralised network model is shown in Fig. 2.1(a). The centralised network architecture comprises of a number of nodes centralized around a single central base station. The distributed architectural model is shown in Fig. 2.1(b). It has no central base station. Instead, each node shares information with its neighbours and is responsible for its own information processing.

In the centralized network architecture the SU base station (BS) handles all the spectrum sensing, allocation and management of all SU nodes [20]. This architecture has a number of advantages. The first advantage of the centralized architecture is that it takes the computational load off individual CR nodes. This results in reduced complexity of the SU nodes and means that they are cheaper and simpler to build. It also reduces overhead for SU, which helps them conserve battery power. This is especially useful because CR devices have to be compact and power efficient. A disadvantage of the centralised architecture is that all secondary nodes on the network are entirely dependent on the central base station. If it were to fail the entire network would collapse.

The distributed network architecture relies on cooperation between CR nodes. This type of network is sometimes called the Ad-Hoc network and works with no pre-existing architecture. In the distributed network there is no central node to manage services. In the distributed architecture this job is allocated to each individual node. This of course



(a) Centralised. (b) Distributed.

Figure 2.1: CR Architecture Models

has its advantages and disadvantages, since there is no central node the network becomes more flexible. If any node on the network were to malfunction for any reason the network would not be affected to a large degree. A disadvantage of the distributed network architecture is that it introduces added complexity and overhead for each secondary user on the network. The reason for this is that all calculations and resource management now fall to individual users on the network [20].

2.2 Cognition Cycle

A primary objective of cognitive radios is to perceive the environment that it is operating in and learn from events that occur to generate plans for future action [21][22]. A typical cognitive cognition cycle is shown in Fig (2.2), it consists of four phases each essential to the efficient function of the cognitive radio. The four phases are sensing, analysis, decision and the adaptation. Each phase is essentially dependent on the results of the previous phase, if data corruption by a malicious user occurs during the sensing phase it is easy to see that the entire cycle will be effected. It is essential that a cognitive radio node is able to continually monitor the environment and be able to dynamically configure its operation to take advantage of changes in the radio environment.

The focal aspects of cognitive radio is its ability to sense the surrounding environment, analyse the information that is being apprehended, and use this information to

make a decision on whether the radio frequency band that has been found is the best transmission strategy at that point in time or whether a CR should continue spectrum sensing. [23].

Figure 2.2: Functional Architecture of a Cognitive Radio

The first step in the cognition cycle is the sensing phase. During the spectrum sensing phase a cognitive radio scans through the frequency spectrum until it has found a vacant band that it deems to be suitable for transmission. This is perhaps the most important function of a cognitive radio. The ability to sense spectrum holes by interacting/monitoring its surrounding environment allows it to make the most beneficial decision available. At this stage of cognitive radio operation it is crucial that sensing is done in real time and that the scanning time is as fast as possible. This means that it is essential that the hardware and software components of the cognitive radio are able to modify processing, frequency and bandwidth to enable fast detection of spectrum holes[23]. At the same time it is important that the sensing phase retrieves reliable and accurate information. If a SU is fooled by a malicious user or the sensing phase retrieves a spectrum band that is being used by a primary user then the cognitive radio is in danger causing serious interference to licensed users. A number of spectrum sensing techniques currently exist. These can be broadly classified into three categories: energy detection, matched filter detection and cyclostationary detection. These three techniques will be discussed in more detail in the next section.

The next phase of the cognitive radio cycle is the analysis phase. The analysis phase is responsible for analysis of the information that was acquired in the sensing phase [23]. During the analysis phase possible spectrum opportunities are evaluated and analysed for potential use by a CR. A spectral opportunity is conventionally defined as a band of frequencies that are not being used by the primary user of that band at a particular time

in a particular geographic area [24]. The definition of a spectrum opportunity can be further expanded to include opportunities when a SU is able to transmit at the same time as the PU without causing interference to the PU (or more precisely causing negligible interference). The advancement of MIMO(multiple input multiple output) technology allows us to use directional antennas and beam forming to efficiently transmit information over bands where a primary user is active with very little or no interference to the primary user operation[25].

The next phase in the cognition cycle is the decision phase where a cognitive radio decides which band it should use for transmission. This phase is heavily dependent on the previous two phases and is a direct result of the analysis phase. The decision at the decision phase represents all the information that has been gathered by the spectrum sensing phase and that has been analysed in the analysis phase. The information obtained in the previous two phases insures that the decision that is made achieves the best possible outcomes for the SU. The decision phase is also responsible for defining the parameters for the upcoming transmission [23]. Parameters that are defined in the decision phase include the transmission power, the transmission start time, modulation rate and number of antennas to be used [23][24][26].

The fourth and final phase is the adaptation phase. During this phase the information and decisions that were accumulated in the previous phases are executed by the CR. The adaption phase is a direct extension of the decision phase and is where all the parameters from the previous stage are implemented. At the conclusion of the adaptation phase the cycle reverts back to the sensing phase where the cognitive radio monitors the radio frequency spectrum looking for spectrum holes. Every time the cognitive radio cycles through the cognition cycle it continually adapts and learns from its environment. The operation of the cognitive radio becomes more sophisticated with the conclusion of each cycle [23] enabling it to make better decisions and attain better results.

2.3 Spectrum Sensing

The most important aspect of cognitive radio operation is efficient and accurate spectrum sensing. Spectrum sensing is defined as the task of obtaining awareness about the spectrum usage and existence of primary users on a specific frequency band [8]. Spectrum sensing allows a secondary user to identify frequency bands that are not being utilized by PUs. The sensed spectrum bands can be classified into three categories: black spaces, white spaces and grey spaces [27]. White spaces correspond to spectrum bands that are completely vacant. Gray spaces are partially used spectrum bands that can be considered by secondary users, they are bands that are partially occupied by low power PUs. The black spaces are spectrum bands that are occupied by primary users and should not be considered by SUs. There are a number of different spectrum sensing techniques available for cognitive radio. The goal of spectrum sensing is to decide between two hypotheses[27].

$$x(t) = \begin{cases} n(t) & H_0 \\ hs(t) + n(t) & H_1, \end{cases}$$

where, the H_0 hypothesis specifies that no primary user was sensed on the channel. The H_1 hypothesis indicates that a primary user is currently occupying the channel. s(t) denotes the signal from the primary user, $n(t)$ is the channel noise(AWGN) and h is the shadowing constant. The first and simplest technique for spectrum sensing is called energy detection. This basic technique measures the received signal strength of the incoming signal. In energy detection the signal is measured over a period of time and the average signal strength is acquired. This average is then compared to a pre-set threshold which determines if the transmitted signal is a primary user or just noise. A key feature of energy detection is that it does not require any knowledge about the characteristics of a primary user signal. This enables energy detection to determine if the channel is being used very quickly. Figure 2.3 provides a summary of the different sensing methods with their corresponding sensing accuracies and complexity.

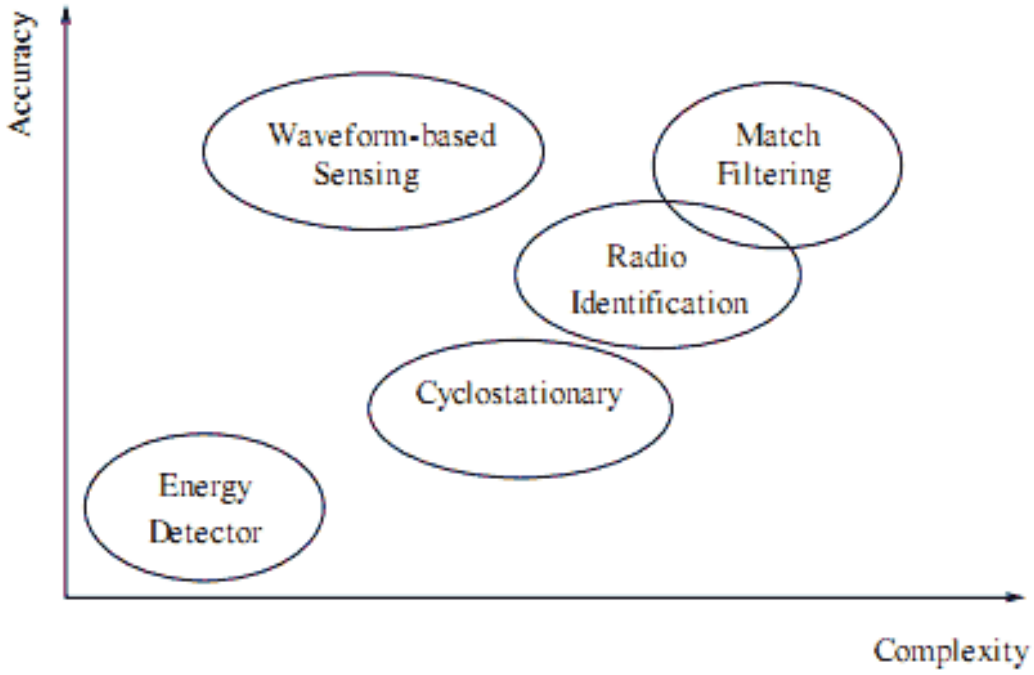


Figure 2.3: Spectrum sensing techniques.

2.3.1 Energy detection

Energy detection is the primary means of spectrum sensing when the secondary user has no prior knowledge about the signal characteristics of the primary user. Energy

detection is very simple to implement and does not require complicated hardware for implementation. Energy detection has a number of drawbacks. It is not able to distinguish between channel noise and the signal from a transmitter. This means that noise has great effect on its performance. It has been shown, that energy detection performs badly in low noise environments [28] [29] [30]. Another problem with energy detection is the threshold selection. It is very difficult to set a threshold that will optimize performance.

2.3.2 Matched Filter Detection

Matched filter detection is a more sophisticated form of detection then energy detection. In matched filter detection the incoming signal from the primary user is put through a filter and is correlated to a signal sample. The result of the correlation is compared to a predefined threshold and a decision is made on whether the signal came from a primary user or not [31]. Matched filter detection performs much better than energy detection. It is able detect a primary user more accurately than energy detection and is much less susceptible to noise than energy detection. Matched filter detection has also been shown to be very quick and efficient [18]. Its main disadvantage is that in order to work it must have prior knowledge of the PUs signal. If it does not have this its performance is very poor. Therefore, even with its improved performance over energy detection it is often overlooked for energy detection because of its dependence on prior knowledge [32].

2.3.3 Cyclostationary Detection

The idea of the Cyclostationary feature detection is to utilize the built-in periodicity of a modulated signal [33][27]. Cyclostationary feature detection works by auto correlating the incoming signal which separates the signal from the noise. The fact that noise on the channel is not periodic in any way allows Cyclostationary to efficiently separate it from the signal. This means that unlike energy detection Cyclostationary is not affected by noise. Cyclostationary is also able to distinguish between a secondary user signal and primary user signals. The reason for this is that different wireless systems usually employ different signal structures and parameters [18]. Cyclostationary requires that the incoming signal has Cyclostationary properties, it also requires the value of a cycle frequency [27]. A major disadvantage of Cyclostationary is that it needs complicated equipment for implementation, it also needs multiple FFT calculations which make it slow and computationally expensive to implement [33][27]. Cyclostationary has been shown to be much more accurate than energy detection and because of its high tolerance to noise on the channel it provides much better results in noisy environments. However, compared to energy detection it is much slower and much more expensive to implement.

2.3.4 Waveform based sensing and radio identification based sensing

In addition to the classic spectrum sensing techniques for cognitive radios we present two additional techniques: waveform based sensing and radio identification sensing. Waveform based sensing takes advantage of patterns in the preamble and pilots of a transmitted signal to identify a primary user. A preamble is a sequence of bits transmitted before each signal burst. If a secondary user on the network has knowledge of what patterns are used by a primary user they can analyse the preamble and decide whether the signal is coming from a primary user or not[34]. Radio identification based sensing uses priory knowledge about the transition techniques used by the primary user. This allows a cognitive radio to identify key features about the primary user which help it detect the presents of a PU on the spectrum band [34].

2.3.5 Combined Detection

Benko [1] present the idea of using a combination of these techniques to achieve better results than each individual technique could achieve by itself. In [1] Benko proposes an algorithm based on a combination of energy detection and feature detection. The method proposes to use energy detection to find candidates and feature detection to identify the type of signal on the band. In the first part of the technique large parts of the spectrum are sensed using energy detection, at this stage the sensing sensitivity is not important. After the energy detection scheme identifies possible bands for use. Feature detection is used to determine with higher accuracy if a primary user signal is present or not [1][33]. The use of a combination of sensing techniques helps improve both accuracy and speed of sensing. Energy detection is used to scan a large number of frequency bands very quickly. Then the most promising bands are selected and are further scanned using feature detection to increase the accuracy of the result.

2.3.6 Challenges

The primary requirement of spectrum sensing is that the detection is fast and accurate. A successful spectrum sensing algorithm must achieve an optimum balance between speed and accuracy. It is important that sensing is done as fast as possible in order to increase transmission time. However, if sensing is done fast but at a low accuracy the overall performance of the network will decrease. Multipath fading and dispersion can cause serious degradation to signals in wireless networks. These are major challenges that secondary user networks need to overcome in order to be able to accurately and reliably sense the presents of a primary user on the network. The location of the cognitive user network can have a large effect on the amount of noise and interference that a signal is subject to. This means that spectrum sensing techniques must be flexible and must be able to deal with noisy environments. Another major challenge of spectrum sensing is the implementation of the right detection method for the right application. In areas where

there are large amounts of noise, energy detection is not going to be very effective as a solution. If secondary users have prior knowledge of the primary user signal then matched filter detection is the optimum solution. Spectrum sensing is still an open research field and an optimum method is yet to be found.

2.4 Software Defined Radio (SDR)

Software defined radios (SDRs) are an essential part of CR implementation. Cognitive radio devices are designed to be highly adaptable radios that are able to change their functionality to suit changes in the environment. It is therefore essential that cognitive radios have an extremely adaptive and flexible software and hardware platform. In [35] Software defined radios are defined as "a collection of hardware and software technologies where some or all of the radios operating functions (also referred to as physical layer processing) are implemented through modifiable software or firmware operating on programmable processing technologies. These devices include field programmable gate arrays (FPGA), digital signal processors (DSP), general purpose processors (GPP), programmable System on Chip (SoC) and other application specific programmable processors.

There are a large number of models that describe both cognitive radio and software defined radios. Figure 2.4 illustrates a model that relates cognitive radios to a software defined radios. This model enforces the important relationship that must exist between cognitive radios and SDRs. A cognitive radio aims to satisfy the radio link requirements of users [22]. It does this by continually monitoring the environment using a number of sensors, its goal is to be agile and be aware of changes in the environment as quickly as possible. After measurements are taken, they are analysed and evaluated by the cognitive radio device. If sufficient change has occurred in the environment the cognitive radio engine will implement changes to insure the required level of performance is maintained, it is able to implement the changes by modifying the SDR framework. This insures that the upper layer functionality requirements are met.

There are a number of advantages of implementing cognitive radio devices using software defined radios. Low cost manufacturing, implementation and maintenance of a product are key considerations during the development of any product. The use of software defined radios will enable development companies to implement a low cost high quality cognitive radio device. SDRs not only offer low cost manufacturing but also low cost of upgrading. Software defined radios allow for remote troubleshooting and reprogramming which decreases the cost of maintenance and error correction within a CR device.

A key feature of SDRs is its ability to offer great power efficiency for cognitive radio nodes. Power efficiency is an essential feature in cognitive radio design since most cognitive radios are going to be implemented in mobile devices such as mobile phones.

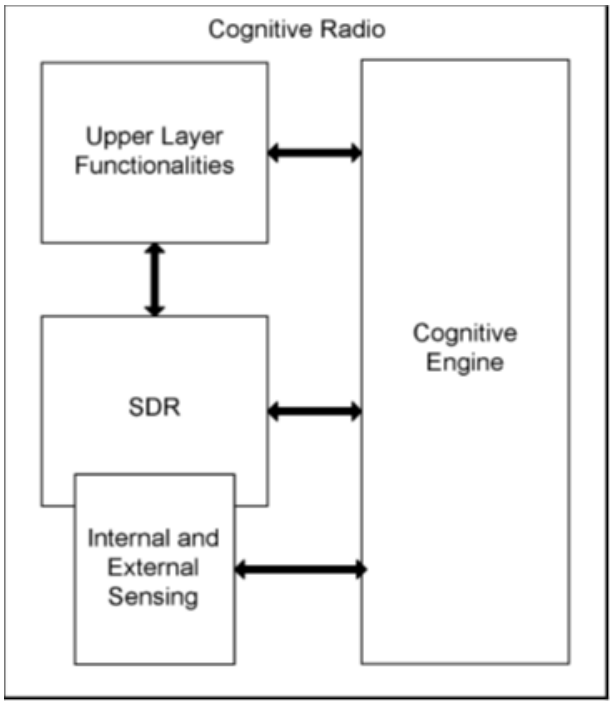


Figure 2.4: Shows the relationship between SDR and CR

These mobile devices have certain design restrictions implemented on their size. To insure that this restriction is kept manufacturers must use appropriate components. This has a great impact on the battery life of the device and means that devices have to be built to be as power efficient as possible.

A simple architectural model of a SDR is shown in figure 2.5 [22]. This model is made up of three different parts: the configurable digital antenna, the software tuneable analog radio and an Impedance Synthesizer. All three components of a software defined radio are fully reconfigurable, which allows the device that is using the SDR to be flexible. The reconfigurable digital radio performs digital radio functionality, the software tuneable analog radio performs functions that are associated with analog radio functionality and the impedance synthesizer is used to optimize the performance of software tuneable antenna systems [22].

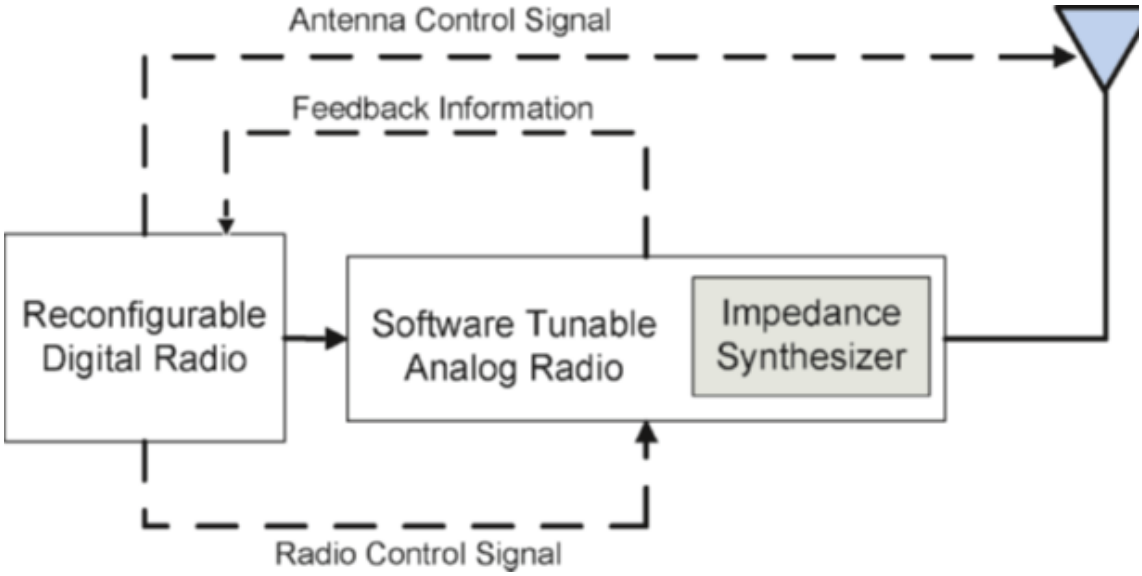


Figure 2.5: Ideal SDR architecture.

2.5 Secondary User Cooperation

Cooperative communication allows users in a wireless network to share resources and create collaboration through distributed transmission and processing [18]. Cooperation is an essential part of cognitive radio development. It is important that secondary users in the network are able to share information about network conditions, spectrum availability and the presents of malicious users. Cooperation promises significant capacity and multiplexing gains in CR users. It also realizes a new form of space diversity to combat the detrimental effects of severe fading [18].

There are two approaches to cooperation in the cognitive radio framework: distributed cooperation and centralized cooperation. The choice of which approach to use depends heavily on the type of architecture that is going to be implemented as well as the application of the system. The centralized cooperation approach uses a central node called a master node to regulate and collect information from all the other secondary users. This node is responsible for collecting and processing information from each secondary user on the network. A common application of centralised cooperation in cognitive radios is spectrum sensing. A single user is not able to scan large portions of the radio frequency spectrum. However, when a large number of secondary users cooperates, they are able to scan a much larger number of channels. In this instance each node in the network scans a part of the spectrum and sends its findings to the master node. The master node collects this information, processes it, and then allocates available channels to secondary users upon request [18][18].

An advantage of the centralized approach to cooperation is that the secondary users

on the network can be very simple devices. They do not need to have large processors or waste a large amount of power collecting and processing information, they simply relay information to the master node to process and regulate in a fair way. A disadvantage of this approach is its centralization of resources. If the master node is attacked or corrupted or if it malfunctions in any way the consequences for the secondary users on the network could be severe [36].

The distributed approach is a decentralized approach in which each secondary user is responsible for collecting and processing information from its neighbours. In the distributed approach there is no master node to regulate information instead each user processes its own information and makes decisions on which action it should take based on information from other secondary users on the network. The distributed approach requires a much higher level of autonomy from each individual user. Each user has to have the processing ability and power to be able to process all the information that it is receiving [37]. An advantage of the distributed approach is robustness against malfunction and attacks. The distributed architecture of this approach means that if any single user is corrupted or the network will still function with fairly high efficiency [18].

Cooperation between nodes allows cognitive radio networks to perform more efficiently. As an example, using cooperation in spectrum sensing allows for a reduced number of false alarms and miss-detections. The reason for this is that it is very unlikely that all the users on the network are going to report false information, if a small group of users are sending false information they can easily be ignored. Cooperation insures that secondary users are able to sense the frequency spectrum more efficiently and with a much higher degree of accuracy. Cooperation allows cognitive radio networks to utilize much more spectrum then with conventional approaches.

The implementation of cooperation between secondary users means that a control channel must be established so that messages between secondary users can be transmitted. The allocation of the control channel means that a portion of the total bandwidth must be assigned to the control channel and can no longer be used by secondary user to transmit data. In cooperate spectrum sensing we must also insure that there is synchronization between secondary users. It is important that during the spectrum sensing phase there are no secondary users transmitting[18].

Chapter 3

Cognitive Radio Security

3.1 CR Security Overview

The nature of cognitive radio networks introduces an entire new suite of threats that are not easily mitigated [38]. Cognitive radio operation is based on sensing, analysing and learning from the environment. Cognitive radios awareness, learning and analytical capabilities are important features which are seen as its greatest advantage over conventional wireless networks. However, these advantages can be used as possible points of infiltration by malicious users looking to disrupt network operation to gain an unfair advantage. In a CR network cognitive radios share information with their neighbours in order to make better informed decisions. This presents a possible infiltration point by which an attacker can cause both short and long term damage to a CR network. A simple attack on a secondary user can be propagated quickly and have far reaching effects on the entire network. Hence, it is important that appropriate counter measures are put into place to insure that malicious users are not able to affect network performance and behaviour through simple spectral manipulation [38].

The information sensed by a cognitive radio is used to construct a perceived environment that will impact, in a certain way, on the current and future behaviours of all the nodes in the network. This means that false or corrupted information can have both short and long term effects and can decrease network efficiency significantly. The fact that secondary users are constantly communicating with each other means that an attack only has to target one node to have a long lasting and far reaching effect on the network. This makes attacks on cognitive radio networks simple to deploy, but hard to counter. A number of prominent features of cognitive radios can be infiltrated by malicious users to cause significant damage to not only a single radio but the entire network [39], these include:

- Cognitive radios awareness of the surrounding environment and internal state: This presents a great opportunity to cause havoc on the network by sending false information to the secondary user creating a misconception about its operating environment.

- Cognitive radios ability to adapt to its environment to meet requirements and goals: Can be used to effect fundamental belief of a secondary user and insure that a secondary user takes actions that benefit the attacker
- Cognitive radios ability to Learn from previous experiences to recognise conditions and enable faster reaction times: this allows a malicious user to create a long lasting impact on the cognitive radio network with small continues attacks.
- Cognitive radios ability to anticipate events in support of future decisions: Once again this allows for long lasting impact on the networks with very little effort required from the attacker.
- Cognitive radios collaboration with other devices to make decisions based on collective observations and knowledge: This enable the attacks to have a far reaching effect on the network by simply attacking a single user. This enables the attack to use minimum resources and cause maximum damage.
- - Cognitive radios wireless communication capability: Provides an opportunity for malicious users to intercept information by eavesdropping.

Interestingly, the potentially exploited features of cognitive radios are also the ones that are used to mitigate many of the attacks that CR network face, these counter measures are listed as follows:

- The ability to collaborate: this helps secondary users in the network identify an attacker by sharing information and findings. It also helps spread findings around the network to insure that other secondary users are not affected by the same attacker.
- The ability to learn: CRs ability to learn insures that it does not make the same mistakes by learning from previous attacks.
- The ability to anticipate: It is able to anticipate attacks by the anticipating behaviour of the malicious user.

In order for cognitive radio deployment to be successful it is important to insure that it is well protected against security threats. It is therefore critical, that effective techniques are developed to insure that cognitive radio users are able to utilise their device to its full potential. In order to develop effective techniques it is essential that a CR security framework is established that identifies as many different types of attacks as possible [40]. The current taxonomy of attacks is outlined in the next section.

3.2 CR Security Framework

Research into cognitive radio security has led to a classification of a number of security threats to a cognitive radio network. these can be grouped into the following classes:

primary user emulation attacks, spectrum sensing data falsification attacks, MAC layer attacks, network layer attacks, cross layer attacks and SDR attacks. This section provides a broad overview of each of these attacks.

3.2.1 Primary User Emulation Attacks(PUEA)

A fundamental principle of cognitive radio is to allow a secondary user to access the radio frequency spectrum when primary users are not using it [39]. The secondary user is allowed to use a specific band provided that it vacates the band as soon as the primary user becomes active. If a secondary user detects another secondary user using the band, mechanisms should be in place to insure that the spectrum is shared fairly [41].

This inherent feature of CR networks presents a target for malicious users to exploit. If a malicious user was able to mimic the signal characteristics of an incumbent primary user, they could trick secondary users into thinking that the channel is occupied. This would result in SUs immediately vacating the channel and leaving the entire channel vacant for a malicious user to use uncontested. Primary user emulation attacks are classified into two groups:

- Selfish nodes: the selfish node mimics the characteristics of the primary user to increase their own share of the spectrum. Their main goal is not to attack the network or other users it is simply to gain an advantage and uncontested spectrum.
- Malicious node: the malicious node mimics the characteristics of a primary user to cause denial of service attacks (DOS). These nodes are not interesting in increasing their resources but in denying resources to other users. Malicious nodes use spamming attacks on multiple spectrum bands to insure that secondary users cannot transmit on as many channels as possible.

Primary user emulation attacks have both long and short term effects on the CRNs. The short term effects on secondary users are the inability to access a channel as long as the malicious user is transmitting their signal. Another short term effect of a PUEA is that they make secondary users constantly switch from one band to another this causes extra overhead in setup costs associated with synchronisation and handover. Cognitive radios have the ability to learn and anticipate when spectrum bands are going to be available based on previous experience. Therefore, an attack on multiple bands can results in secondary users staying away from these bands in the future casing long term damage and degradation of bandwidth.

A more sophisticated form of primary user emulation attack is possible if the malicious user has knowledge of how the CR network operates [41]. In a cognitive radio network a period of operation called the quite time is dedicated to spectrum sensing, during this time all secondary users are idle. This period presents the perfect opportunity for malicious users to strike. If an attacker is aware of when the quite times are going

to take place, they are able to save resources and attack when the opportunity presents itself. This insures that they cause the most amount of damage with the least amount of effort. There are a number of different techniques that mitigate the effects of primary user attacks. In this paper we present a belief propagation algorithm that is extremely effective in moderating PUE attacks.

3.2.2 Spectrum Sensing Data Falsification Attacks(SSDFA)

Spectrum sensing data falsification attacks occur when a secondary user sends out false information about its spectrum sensing results [40][39]. This can cause other users on the network to miss detect an active primary user, causing interference. The attack can also trick users into thinking that a primary user is active when they are not, to gain uncontested access to the frequency band. Spectrum sensing data falsification attacks use the same trick as PUEAs to reduce network efficiency by making secondary users think that spectrum bands are being utilized when in fact they are idle. Similar to primary user emulation attacks spectrum sensing data falsification attacks can be characterised into three groups:

- Malicious User: the malicious user intentionally sends out falsified information so trick the secondary user into thinking the PU is idle when they are not, or that the PU is active when they are not. In the first case the goal of the malicious user is to cause interference to the primary user. In the second case the goal of the malicious user is to insure that the secondary user does not use the available spectrum.

- Greedy User: the greedy user continuously reports that a primary user is active when they are not [39]. This insures that other secondary users on the network do not use the band and allows the greedy user to use the band uncontested.

- Unintentionally misbehaving user: the unintentionally misbehaving user reports false information about their spectrum sensing results because their software/hardware is malfunctioning. They have neither greedy nor malicious intentions but can still have severe effects on network performance.

The SSDF attack targets both the centralised and distributed topologies [41]. In the centralised topology all secondary users send their sensing information to a central fusion centre where the observations are processed and the BS allocated resources in a fair manner. In a distributed network topology resources are delegated using collaboration and each node is responsible for processing their own sensing information.

A number of techniques exist to combat the effects of spectrum sensing data falsification attacks in CR networks. In [42] a cooperative scheme is proposed, in this scheme each secondary user calculates the probability that the primary user is actively using a channel. Each secondary user sends its findings to the fusion centre where they are added and averaged. If the average value is greater or equal to a predefined threshold the fusion centre concludes that a primary user is active. If it is under the threshold, the fusion

centre concludes that the band is unoccupied. The author in [42] proposes a very conservative threshold of 1, which means that if a single user on the network reports a low probability that the PU is active, the fusion centre will conclude that the band is not free.

In [43] the authors propose a scheme called Weighted Sequential Ratio Test (WSRT) which is a slightly more advanced scheme then the one discussed in [42]. WSRT introduces a reputation value for each node on the network. In WSRT each node on the network is given an initial value of zero for its reputation, for each correct local report that it provides its reputation is increased by 1. When the probability of a primary user is active is evaluated, the evaluation takes into account the reputation of each node in the network. Users with a low reputation contribute less to the final decision then those with a higher reputation which results in more accurate sensing and offers extra protection from malicious users trying to sabotage the network.

Another approach for the mitigation of SSDFA attacks is proposed in [44]. The proposed scheme is based on a trust value assigned to each SU in the network. This scheme assumes that malicious nodes are either an always yes node of an always no node. This scheme relies on pre-filtering of the data to identify and nullify the malicious users [44]. It uses the statistics from a large number of users to identify an attacker very quickly. SSDFA attacks are incredible hard to mitigate, the above algorithms offer ways to reduce the effects of SSDFA attacks but are unable to completely eliminate their effect. This remains an open research area.

3.2.3 MAC Layer Attacks

The MAC layer operates closely to the physical layer to insure that hardware components accomplish their goals. The most important function of the MAC layer is to insure that SUs do not interfere with primary users on the network. A common control channel is used to insure that this does not happen. The common control channel is a separate dedicated pre-defined frequency channel that allows SUs to exchange control information [39]. The MAC layer is responsible for maintaining the control channel. The common control channel is very important. It controls all network operations which makes it an attractive target for malicious users. The three most serious threats to the common control channel are:

- MAC spoofing: where an attacker sends a large amount of messages to try and disrupt channel operation [39].
- Congestion attacks: congestion attacker try and flood the common control channel to cause denial of service attacks.
- Jamming attacks: where attackers cause interference to the physical layer resulting in denial of service.

The common control layer is vital in CR operation. It is therefore very important that it is protected from malicious users because if it is compromised network performance is severely affected.

3.2.4 Network Layer Attacks

Research into cognitive radio technology is primarily focused on the physical and MAC layer protocols. Routing challenges faced in CRNs originate from the need for transparency of CR activities to primary users [41]. This is further increased because CRs must vacate a frequency band immediately if a primary user becomes active. The two main types of network layer attacks are sinkhole attacks and hello flood attacks.

In sinkhole attacks a malicious user advertises himself as the best route to a destination. This entices secondary users to send their traffic through the malicious node enabling him to control how and if packets make it to their destination. With this control of network traffic the malicious user is able to change and modify packets or send false information from one node to another. This attack is most effective when the network is centralised, in this instance the attacker can advertise themselves to be the central node giving them total control of network resources [41].

In a hello flood attack an attacker sends out broadcast messages to all nodes on the network with enough power to convince them that they are their neighbour. This type of attack is designed to disrupt the network. For example, an attacker can advertise that they are the best link to a certain destination but in reality they are very far away from the node. Then, when the node tries to send packets they are lost because of the distance between the nodes [41].

Sink hole attacks are very hard to detect and even harder to mitigate. The author in [41] proposes a method called geographic routing in which protocols construct a topology on demand using only local communications and information without initiation from the base station.

The primary method for mitigating against hello flood attacks is to use symmetric keys. A symmetric key is shared between a secondary user and a trusted base station. When a secondary user wants to transmit to a base station these session keys are used as authentication between the two parties. To insure that an attack does not create their own session keys the base station must limit the number of session keys that are given to secondary users and must suspect nodes that appear to have too many neighbours to be malicious nodes. Alternative methods use authentication and encryption to ensure safe communication between nodes, the session key is preferred because of its low overhead [41].

3.2.5 Cross Layer Attacks

The previously discussed attacks were all focus on a single layer. The PUAE and the SSDF were focused on the physical layer, the hello flood attacks and sinkhole attacks focus on the network layer. However, it is possible to launch attacks on more than one layer at the same time. One of the most popular types of attacks is to use the PUEA (physical layer) to disrupt the Transmission Control Protocol (TCP) (Transport layer) connection. In this attack the attacker uses a PUEA to insure that secondary users perform frequency handoffs. When this handoff takes place, the TCP will not be aware of the handoff and will keep creating logical connections and sending packets without receiving acknowledgments [41]. This type of attack is called the lion attack and it results in loss of packets and large delays for secondary users.

To mitigate against the lion attack, [42] suggest that an algorithm is developed that allows sharing between the physical, data link and transport layers. This insures that if a secondary user has to vacate a frequency band the TCP protocol will be aware of the change. To stop the threat of eavesdropping [42] proposes a scheme called group key management that allows secondary users to encrypt, decrypt and authenticate themselves. In [39] the author suggests that in order to combat cross layer attacks it is essential that cognitive user have a degree of common sense.

3.2.6 Software Defined Radio Security

Software defined radios are critical in insuring that cognitive radios are able to function as they were intended. SDRs provide flexible software and hardware components so that a cognitive radio is able to make changes to its operation to reflect changes in its environment. It is therefore very important to establish security measures that insure that software defined radios and not compromised in any way [41][40]. There are two types of attacks that effect software defined radios: software based attacks and hardware based attacks.

CRs must be flexible. They must be able to update their software regularly from the internet. This presents a target for a potential attack. Software based attacks involve manipulating software components to insure that the CR malfunctions. If the attacker was able to corrupt a software update they would be able to change the operation of a CR to suit their purpose. This could involve reprogramming the CR to send spectrum sensing data falsifications information and effecting how the entire network operates. SDR Hardware attacks are based on software manipulation, in this form of attack the malicious user would modify software components to change the operation of hardware components and make them malfunction.

Software-based protection schemes involve the deployment of tamper-resistance techniques to defend against malicious or buggy software installations. These schemes insure

that downloads and distribution of software for the SDR are secure. It ensures that downloads are protected against malicious users who want to corrupt them. Hardware protection schemes include modules that act as isolation layers between hardware and the software components of the SDR [41].

Chapter 4

PUEA mitigation algorithms

4.1 Introduction

This chapter outlines two PUE attack mitigation techniques that were presented in [2]. We present each technique along with their simulation results. Based on these results we draw conclusions about the effectiveness of each technique in the mitigation of PUEAs. In gain a better understanding and to verify the results in [2] we use MATLAB to simulate the various scenarios presented by the authors. The first section provides and overview of MATLAB the simulation language that was used throughout this project. The second section presents the first mitigation technique that uses triangulation to localise and incoming signal. The last section presents a more accurate and effective method based on belief propagation that uses cooperation between secondary users on the network to exchange information about whether a secondary user believes a transmitter is a PU or an attacker.

4.2 MATLAB

The performance assessment of each technique is done using a high level simulation language called MATLAB [11]. MATLAB uses an interactive environment for numerical computation, visualisation and programming. MATLAB simulations were used to evaluate the performance of each technique. The results of each simulated technique will be used to draw a conclusion on whether the technique is suitable for mitigation of primary user emulation attacks.

4.3 Triangulation based mitigation algorithm

This section presents a mitigation technique that is based on a triangulation algorithm presented in [2]. The triangulation technique is a verification scheme that calculates the location of the attacker and compares this location to a known location of a primary user. If the two locations match the suspect is assumed to be a primary user, if they do not

match, the secondary user concludes that the suspect is a malicious user and ignores him completely.

The triangulation technique assumes that a secondary user has a maximum transmission range of about 20 meters, which corresponding to a transmission power of a few watts. Primary users are assumed to be TV towers that have a transmission range of a few km, which corresponds to a few hundred watts. It was assumed that secondary users know the location of the primary users. It was also assumed that the malicious user is equipped with a cognitive radio and is able to change its modulation mode, frequency, location and transmission output power. The mobility and flexibility of the malicious users adds to the complexity of primary user emulation attacks. The details of the protocol for communication between secondary users is beyond the scope of this project, however we assume that information is exchanged without error.

Based on these assumptions, a triangulation scheme is proposed in [2] which works by calculating the location of a PUE attacker using the received signal strength (RSS). We denote SU_1 and SU_2 as secondary user one and secondary user two. The location of a secondary user SU_1 is given by (X_1, Y_1). Both secondary users SU_1 and SU_2 are able to receive signals from both the attacker and the PU. Each secondary user records the RSS of each incoming signal. Then, using cooperation they are able to exchange information about each other location and the RSS measurements that they recorded. RSS_1 denotes the received signal strength at user 1. The author in [2] uses a statistical log loss signal propagation model to model signal propagation and calculate the RSS at each user using the following equation:

$$P_r(dBm) = P_t(dBm) - \alpha log(d) - s, \tag{4.1}$$

where, P_r denotes the RSS measurement obtain at a node in dBm, P_t is the transmission signal strength in dBm, d is the receiver/transmitter distance, α is the path loss constant and s represents the degree of shadow fading. In this instance we assume that secondary users SU_1 and SU_2 are close enough to assume that $s_1 = s_2$. This means that the shadowing constants cancel each other out and we are left with equation 4.2:

$$P_r(dBm) = P_t(dBm) - \alpha log(d) \tag{4.2}$$

Since the secondary users on the network do not know the transmit power of the attacker. The author uses the RSS measurements from equation 4.2 to derive equation 4.3:

$$\eta = \frac{d_{attacker,SU_1}}{d_{attacker,SU_2}} = 10^{\frac{RSS_1 - RSS_2}{\alpha}} \tag{4.3}$$

$d_{attacker,SU_1}$ and $d_{attacker,SU_2}$ represent the distances between the attacker and secondary nodes SU_1 and SU_2. RSS_1 represent the RSS measurements obtained at SU_1 and RSS_2 represent RSS measurements obtained by SU_2. Our objective is to calculate the exact

location of the attacker without knowledge of the transmit power of the attacker. If we denote (x, y) as the location coordinated of the attacker, we obtain the following equation:

$$\eta\sqrt{(x-x_1)^2-(y-y_1)^2} = \sqrt{(x-x_2)^2-(y-y_2)^2}, \tag{4.4}$$

where, (x_1,y_1) denote the location of SU_1, similarly (x_2,y_2) denote the location of SU_2. From this we can calculate the trace of the attacker with a circle of radius and center location:

$$\text{radius} = \frac{\eta d_{12}}{\eta^2-1}, \text{center at } \left(\frac{\eta^2 x_1 - x_2}{\eta^2-1}, \frac{\eta^2 y_1 - y_2}{\eta^2-1}\right), \tag{4.5}$$

where, d_{12} represents the distance between SU_1 and SU_2. The proposed triangulation technique uses a reference node which in this case is node SU_1. Node SU_1 interacts with three other users SU_2, SU_3 and SU_4 and creates a circle with each SU. The formation of the circles is illustrated in Fig (4.1). Note that if $\eta = 1$ we are left with a straight line that is orthogonal to the straight line (SU_1SU_2) that passes through the midpoint between SU_1 and SU_2. For this application we consider the cases when $\eta \neq 1$. It is also important to note that if $\eta < 1$ we have a negative radius which is not possible. This fact is not mentioned in [2] but is essential in simulations. To fix this problem we modified Eq.(4.5) to obtain the following:

$$\text{radius} = \frac{\mid \eta d_{12} \mid}{\eta^2-1}, \text{center at } \left(\frac{\eta^2 x_1 - x_2}{\eta^2-1}, \frac{\eta^2 y_1 - y_2}{\eta^2-1}\right) \tag{4.6}$$

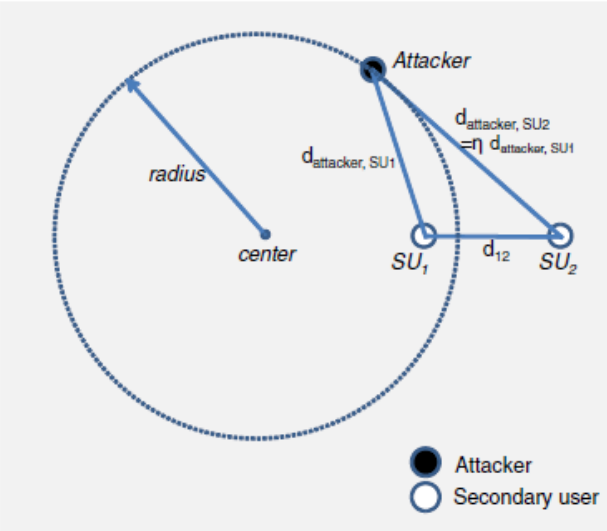


Figure 4.1: Illustration of the formation of the circle.

Fig. (4.2) presents the results of a simulation that was run by the author. In this simulation the author uses four secondary nodes at locations ((400m, 300m), (600m, 300m), (350m, 480m), and (650m450m). These secondary users are deployed in an area that is 700m by 700m. The primary user is not shown in the figure but is placed at (600m, 100m). The attacker is placed where the three circles interact at location (472m, 371m).

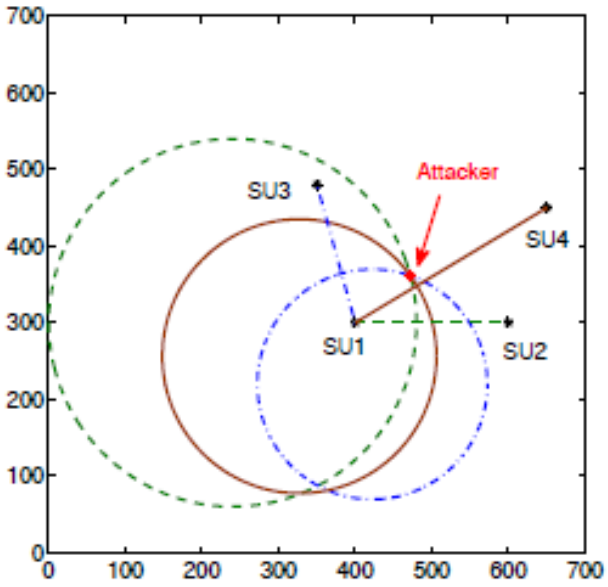


Figure 4.2: Simulation results of the proposed triangulation method.

In simulations the secondary user SU_1 is used as the reference node. Secondary users SU_2, SU_3 and SU_4 are used to calculate the three circles shown in figure 4.2. Each SU calculates their RSS independently. Then secondary user SU_1 interacts with each user to calculate three values of η using Eq. (4.3). These values are used to calculate the radius and centres of the circles using equation 4.6. Using these parameters the author is able to draw three circles that intersect at the location of the attacker. It is important to note that secondary users SU_2, SU_3 and SU_4 do not have to be in each others transmission range for this method to work, they do however have to be in range of SU_1. After the location of the attacker is acquired secondary user SU_1 compares the location of the attacker to the known location of the primary user. If the location of the attacker is the same as the location of the primary user, secondary user SU_1 concludes that the suspect is a primary user. If the two locations do not match, secondary user SU_1 concludes that the suspect is an attacker.

Using MATLAB we were able to replicate the results that were obtained in [2]. The code that was used to simulate this technique can be found in appendix A. The results of the simulations are plotted in Fig (4.3). Comparing the results that are presented by the author in Fig. (4.2) and our simulation results we can see that there is a very close correlation between the two.

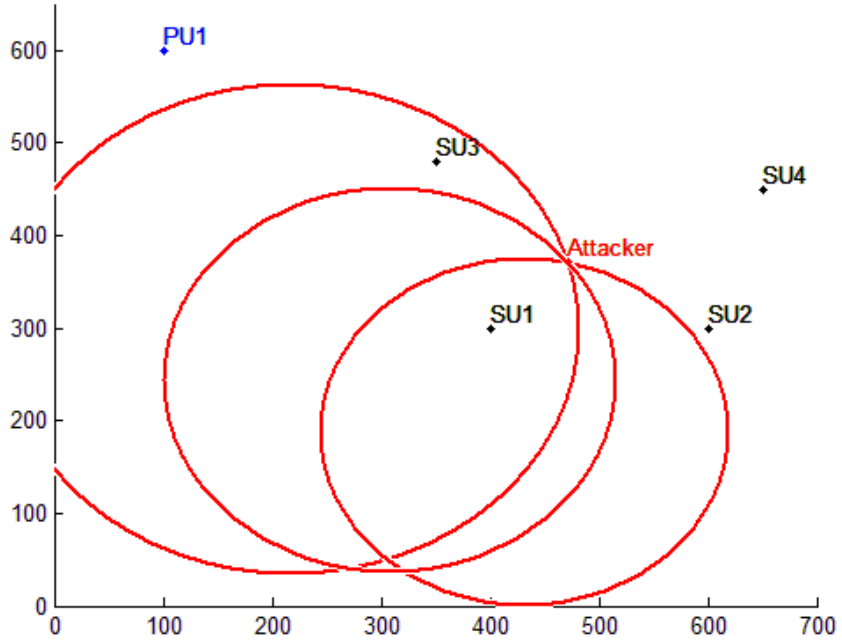


Figure 4.3: Simulation Results for the Triangulation technique.

The simplicity of the technique makes it a perfect entry point into PUAE mitigation techniques. However, the lack of channel degradation factors considered with this technique make it unrealistic for practical implementation. When shadowing is introduced into the algorithm the resulting circles do not intersect at a single point. Fig. (4.4) and Fig. (4.5) show the results that were obtained when independent shadowing for each secondary user is introduced. Fig. (4.4) shows the results of introducing a log normal shadowing constant with zero mean and 0.1 variance. Fig. (4.5) shows the results that were obtained with the introduction of a log normal shadowing constant with zero mean and 1 variance.

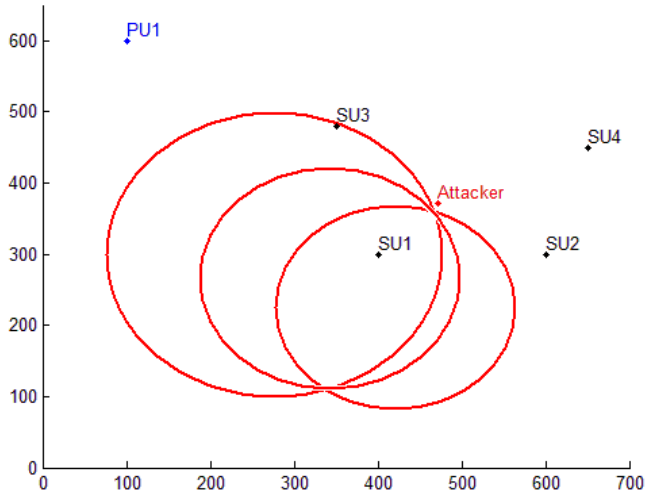


Figure 4.4: Simulation results obtained with shadowing variance = 0.1 and mean = 0.

From simple analysis of the results it is clear that when shadowing is introduced into the algorithm the results that are returned do not offer any useful information about the location of the attacker. It is also evident that the results degrade very quickly as the variance of the shadowing is increased. In Fig. (4.4) we introduced a very small shadowing constant which gave us a small degree of error. Predictably, when we increased the variance to 1 the error in the results also increases leading to meaningless results. On top of its inability to account for shadow fading, for the triangulation technique to work SU_1 needs at least three other secondary users to be in its transmission range to allow it to locate an attacker accurately. This presents an obvious problem when CR users are in isolated situations. The next section introduces a technique called belief propagation that takes into account shadow fading.

4.4 Belief propagation based mitigation algorithm

This section presents a mitigation technique that is based on a belief propagation algorithm presented in [2]. Unlike the previous algorithm belief propagation takes into

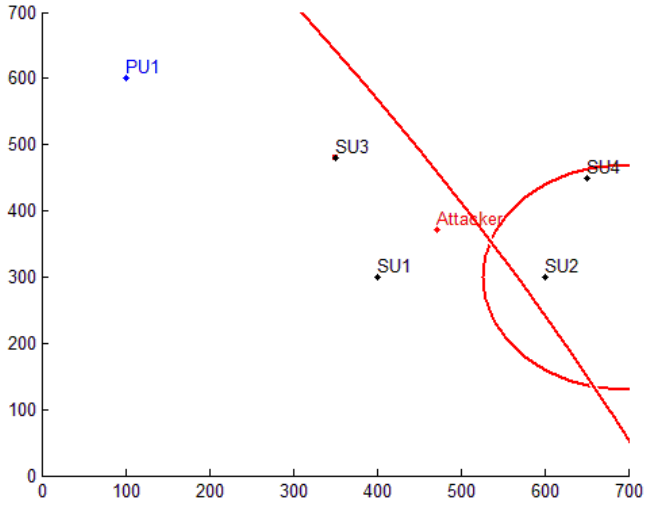


Figure 4.5: Simulation results obtained with shadowing variance = 1 and mean = 0

account shadow fading between secondary users and transmitters. Belief propagation is a cooperative technique that attains better results as more secondary users are added to the network. However, unlike the previous technique it is able to distinguish between an attacker and a primary user using only local observation. When a suspect is identified as a malicious user, a message is broadcast to all SUs on the network informing them that the suspect is a malicious user and that it is to be ignored.

4.4.1 System model

In this section, we describe the basic system model that is used throughout this paper. To model the relationship between the transmit signal power and the received signal power, the author in [2] considers both path loss and log normal shadowing of the channel. Using these assumptions, we define an equation for the received signal strength from a primary user k as:

$$P_{r(PU_k)} = P_{t(PU_k)} d_{PU_k}^{-\alpha} h, \tag{4.7}$$

where, $P_{r(PU_k)}$ represents the received signal power from primary user k, $P_{t(PU_k)}$ represents the transmit power of the primary user k, d_{PU_k} represents the distance between a secondary user and a primary user k, h is the shadow fading constant defined as $h = e^{ab}$ where $a = \frac{ln10}{10}$, b is defined as a random Gaussian variable with a mean 0 and variance σ^2, and α is a propagation loss exponent. From Eq. (4.7) we are able to derive a similar equation to define the received signal power from an attacker as:

$$P_{r(attacker)} = P_{t(attacker)} d_{attacker}^{-\alpha} h_{attacker}, \tag{4.8}$$

where, $P_{r(attacker)}$ represents the received signal power from the attacker, $P_{t(attacker)}$ represents the transmit power of the attacker, $d_{attacker}$ represents the distance between the

attacker and a secondary node and $h_{attacker}$ is a shadowing constant similar to the one used in equation Eq. (4.7).

4.4.2 The Belief Propagation Algorithm

Belief propagation provides high accuracy detection of primary user emulation attacks. In belief propagation, each secondary user performs local observations and calculates the probability that an incoming signal belongs to a primary user. To accurately detect the presents of a malicious user, neighbouring nodes must communicate with each other and exchange local observations. Local observations are exchanged in the form of messages. Each secondary user computes a belief about whether the suspect is a primary user or an attacker according to its own local observations and the sum of all incoming messages from all its neighbours. A final belief is calculated using the sum of all beliefs of all SUs. This final belief is compared to a predetermined threshold. If the final belief is above the threshold, the suspect is deemed to be a primary user. If it is below, the suspect is considered to be a malicious user. The belief propagation framework is based on pairwise Markov Random fields (MRF)[45].

Relative power observations of secondary users represent a pattern of receive powers generated by the location of the transmit station. The exchange of information between secondary users enables recognition of patterns for the purposes of determining whether or not the transmission originates at a known primary user location. In MRF we define Y_i as the local observations at secondary user i, and X_i as the state of the suspect observed at user i. If X_i=1 the suspect is a primary user, if X_i=0 the suspect is a malicious user. The local function at user i is defined as $\phi_i(X_i, Y_i)$. The local function represents the observations made by a secondary user i about whether the suspect is a primary user or not. The compatibility function $\psi_{ij}(X_i, Y_j)$ is used to model the relationship between secondary users. The higher the compatibility function between two users is the more relevant the local observations of the two users become to each other. For example, if SU_1 is 1m away from SU_2 and SU_1 is 30m away from SU_3, then local observations that come from SU_2 to SU_1 will contribute more to the final belief of SU_1 then local observations that come from SU_3. The joint probability distribution is calculated as the product of local observations and all the messages coming into a particular node. The standard formula for the joint probability distribution of Pairwise Markov Fields presented in [45] of unknown variable X_i is given by:

$$P(\{X_i\}, \{Y_i\}) = \prod_{i=1}^{I} \phi_i(X_i, Y_i) \prod_{i \neq j} \psi_{ij}(X_i, Y_j) \tag{4.9}$$

where, I corresponds to the number of secondary users in the network. We aim to compute the marginal probability at secondary user i, which we denote as the belief. The belief at a secondary user i is given in equation Eq. (4.10). It is the product of the local function at user i and all messages coming into user i from all the neighbours of i:

$$b_i(X_i) = k\phi_i(X_i, Y_i) \prod_{i \neq j} m_{ij}(X_v) \tag{4.10}$$

Where k is a normalisation constant that insures that the beliefs sum to 1 ($k = \frac{1}{\prod_{i \neq j} m_{ij}(X_v)}$), $\phi_i(X_i, Y_i)$ is the local function at user i and $m_{ij}(X_v)$ is a message that is received by user i from user j. In order to compute the belief at SU the author in [2] introduces a message exchange equation that is used to iteratively update the messages at each secondary user. A message denoted as $m_{i,j}$ can be understood to as a message from secondary user i to secondary user j. In the first iteration the initial value of $m_{i,j} = 0$, in the l_{th} iteration a secondary user i sends a message $m_{ij}^l(X_i)$ which is updated by:

$$m_{ij}^l(X_i) = C \sum_{X_i} \psi_{ij}(X_i, Y_j)\phi_i(X_i, Y_i) \prod_{k \neq ij} m_{ui}^{l-1}(X_i) \tag{4.11}$$

C is another normalisation constant such that $m_{ij}(1) + m_{ij}(0) = 0$, and therefore:

$$C = \frac{1}{\prod_{k \neq ij} m_{ij}^{l-1}(1)(\psi_{ij}(1,0) + \psi_{ij}(1,1))} \tag{4.12}$$

Finally, after all secondary users finish computing their beliefs, these beliefs are added up and averaged to derive a final belief. The final belief is then compared to a predefined threshold. If the final belief is higher than the threshold, the suspect is believed to be a primary user. If the final belief is lower than the threshold the suspect is believed to be a malicious user:

$$\begin{aligned} &Honest, && \frac{1}{M}\sum_{i=1}^{M} b_i \geq b_\tau \\ &Malicious, && \frac{1}{M}\sum_{i=1}^{M} b_i < b_\tau, \end{aligned} \tag{4.13}$$

where, M is the total number of secondary users in the network, $\sum_{i=1}^{M} b_i$ denotes the sum of all the beliefs of all the secondary users on the network and b_τ denotes the pre-set threshold. It is possible that some users would relay false information to other users in the network. However, false information by a small number of nodes would not influence the final belief value significantly.

Local Function

The local function represents the local observations at a single secondary user. Each secondary user calculates its own local function which corresponds to a probability of a suspect being a primary user. To calculate the local function we must compute two

probability density functions (PDFs). The first PDF is computed using the RSS measurements that are acquired from the primary user and is denoted by PDF_{pu}. The second is a PDF that is computed using RSS measurements acquired from the attacker and is denoted by $PDF_{attacker}$. The local function corresponds to the similarity between the two PDFs. If the PDFs are the same the local function returns a probability equal to 1, which indicates that the suspect is transmitting from a primary user location. The further apart the distributions are the lower the local function and the higher the probability that the suspect is an attacker. The received signal from the primary user can be obtained using the following equation:

$$\frac{P_{r1(PU_k)}}{P_{r2(PU_k)}} = \left(\frac{d_{1(PU_k)}}{d_{2(PU_k)}}\right)^{-\alpha} \left(\frac{h_{1(PU_k)}}{h_{2(PU_k)}}\right), \tag{4.14}$$

where, $P_{r1(PU_k)}$ and $P_{r2(PU_k)}$ are the RSS values from a primary user(PU_k) to SU_1 and SU_2, $d_{1(PU_k)}$ and $d_{2(PU_k)}$ are the distances between PU_k and SU_1 and SU_2. $h_{1(PU_k)}$ and $h_{2(PU_k)}$ represent the shadow fading between PU_k and secondary users SU_1 and SU_2. It is assumed that the channel response is a circular Gaussian variable $\mathcal{CN}$(0,1). If we define q as:

$$q = \frac{h_{1(PU_k)}}{h_{2(PU_k)}} \tag{4.15}$$

We can then obtain the probability density function(PDF) of q as follows:

$$\begin{aligned} f_q(q) &= \int_{-\infty}^{\infty} \mid h_2 \mid f_{h_2} h_1(qh_2, h_2) d_{h_2} \\ &= \int_0^{\infty} q \mid h_2 \mid h_2^2 e^{\frac{1}{2}(q^2 h_2^2 + h_2^2)} d(h_2) \\ &= \frac{2q}{(q^2+1)^2} \end{aligned} \tag{4.16}$$

We can then define $B = \left(\frac{d_{1(PU_k)}}{d_{2(PU_k)}}\right)^{-\alpha}$, and obtain the distribution of $\frac{P_{r1(PU_k)}}{P_{r2(PU_k)}}$ as:

$$PDF_{PU_k}(q) = \frac{1}{\mid B \mid} \frac{2\frac{q}{B}}{((\frac{q}{B})^2+1)^2} \tag{4.17}$$

We can also calculate the expectation of q as follows:

$$E(q) = \int_{-\infty}^{\infty} q f_q(q) d_q = \int_0^{\infty} q \frac{2q}{(q^2+1)^2} d(q) = \pi \tag{4.18}$$

In order to define a PDF for an attacker a secondary user need to collect information from one of its neighbouring secondary users. This includes the location of the secondary user as well as its measured RSS value. If we denote $P_{r1(attacker)}$ and $P_{r2(attacker)}$ as the

received signal strength from the attacker to SU_1 and SU_2 respectively, using Eq. (4.16) we obtain an estimation of the value of $\left(\frac{d_{1_{attacker}}}{d_{2_{attacker}}}\right)^{-\alpha}$:

$$A = \left(\frac{d_{1_{attacker}}}{d_{2_{attacker}}}\right)^{-\alpha} = \frac{P_{r1(attacker)}/P_{r2(attacker)}}{\pi} \tag{4.19}$$

Where π is the expectation from Eq. (17). Then using this we can obtain the PDF of the attacker as:

$$PDF_{PU_{attcker}} = \frac{1}{\mid A \mid} \frac{2\frac{q}{A}}{((\frac{q}{A})^2 + 1)^2} \tag{4.20}$$

To compare the two probability functions the author proposed to use the Kullback Leibler distance. The Kullback Leibler distance is defined as:

$$KL_{(PDF_{PU_k}, PDF_{attacker})} = \int_0^\infty PDF_{PU_k} log \frac{PDF_{PU_k}}{PDF_{attacker}} dq \tag{4.21}$$

The KL distance calculates the difference between the two PDF. If the difference between the PDFs is large the KL formula will return a large number if the distance is small the KL formula will return a small number. We obtain the local function using the results of the KL distance, the local function uses an exponential to insure that the final value of the local function is between 0 and 1, this provides us with a reasonable representation of the local function that fits our requirements. Therefore, the local function can be written as follows :

$$\phi = \exp(-\min_k KL_{(PDF_{PU_k}, PDF_{attacker})}) \tag{4.22}$$

The local function returns a probability that a suspect is a primary user. The higher the probability the more likely the suspect is a primary user, the lower the probability the less likely it is that the suspect is a primary user.

Compatibility Function

The compatibility function is essential for cooperation between secondary users. In the belief propagation framework, the compatibility function is a scalar. The higher the compatibility function between two SUs the more relevant the two SUs are to each other. A reasonable compatibility function may be defined by the following expression:

$$\psi_{i,j}(X_i, X_j) = \exp(-C d^{\beta}_{X_i, X_j}), \tag{4.23}$$

where, C and β are constants, d_{X_i, X_j} represent the distances between secondary nodes i and j. SUs that are distant from each other have different constants therefore less weight is put on a distant SU in the evaluation of the belief. This consideration is independent of the statistical distribution of SUs in the CR network. If the distance between a pair of secondary users is large then the compatibility function tends to zero, if the distance between secondary users is small the compatibility function tents to 1. The compatibility

function is used to insure that users that are far away do not have a large contribution to each others beliefs. The reason for this is that secondary users at different locations suffer from different degrees of shadow fading and the further away users are the less likely that their belief will correspond with each other. It also insures that users that are closer to each other have a greater impact on each others belief.

Complete algorithm

The belief propagation algorithm is summarised in table 1. Each secondary user performs measurements to calculate the primary user probability density function and the suspects probability density function. Then in an iterative way each user computes their local and compatibility functions using equations 4.22 and 4.23. Each secondary user then computes and sends messages to all its neighbouring nodes. The last step of the iteration is where the secondary users calculate their belief using their own local observations and the product of all the messages from its neighbours. After a number of iterations the mean of all the beliefs is calculated and compared to a predefined threshold. If the final belief is lower than the threshold the suspect is thought of as an attacker, if the final belief is greater than the threshold the suspect is deemed a primary user. In both cases the final decision is relayed to all secondary users who will either ignore the transmitter (if he is an attacker) or conclude that a primary user is active and look for another band to transmit on.

Algorithm 1 Complete defence strategy against the PUEA using belief propagation

1: Each secondary user performs measurements using Eq. (4.17) and Eq. (4.20)
2: **for** Each iteration **do**
3: Compute the local function using Eq. (4.22) and the compatibility function using Eq. (4.23)
4: Compute messages using Eq. (4.11)
5: Exchange messages with neighbours
6: Compute beliefs using Eq, (4.10)
7 **end for**
8. The PUE attacker is detected according to the mean of all final beliefs based on comparison against threshold.
9. Each SU will be notified about the characteristics of the attacker's signal and ignore them in the future.

The result of the belief propagation based strategy depends on the ability of secondary users to communicate with each other. The more cooperation between secondary users exists the more accurate the results are going to be.

Simulation Results and Analysis

This section presents the results that were obtained in [2] compared to the results obtained in this project. In the simulation and results section the author presents the results of two separate scenarios. The simulation parameters were given as: the pathloss constant α is set to be 2.5, the transmit power of the secondary user is 0.1W (since the malicious user is also using a cognitive radio this is also the transmit power of the malicious user), we assume this corresponds to a transmission range of about 20 meters. There are 30

SUs deployed in a 100m by 100 meter grid. In the first scenario the attacker is located at (50,50), the primary user is located at (5,5). In the second scenario the attacker remains at (50,50) but the primary user is moved to (40,45), Fig. 4.6 illustrates the network model that was used throughout this project.

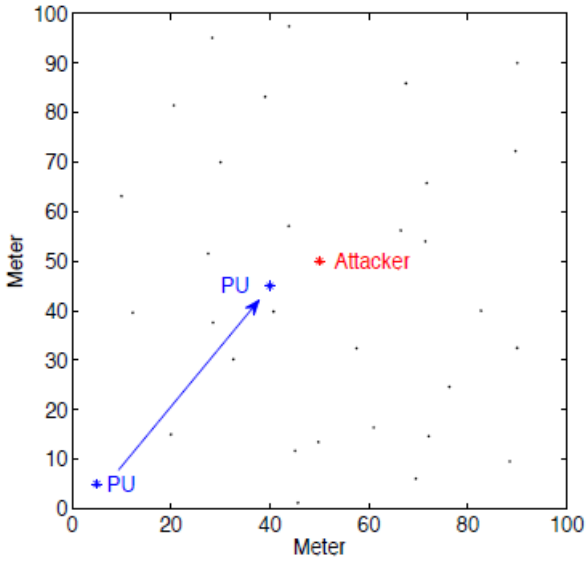


Figure 4.6: Illustration of the CR network.

In the first scenario the attacker is located at (50,50) and the primary user is located at (5,5) the author obtains a final belief value of 0.6. This scenario was run 10 times and the result is the average value that was obtained. The second scenario places the primary user closer to the attacker, when this scenario was simulated an average result of 0.87 was obtained. These results show that as the primary user moves closer to the attacker the final belief increases. When the attacker and the primary user occupy the same position the final belief should be one. However, due to shadow fading the final belief will never be exactly 1(but very close to 1). Fig. 4.7 demonstrates how the final belief varies as the distance between the attacker and the primary user is changed. By observing the graph it is evident that as the distance between the primary user and the attacker increases the mean of the final beliefs decreases.

Our goal was to model the proposed system and use belief propagation to achieve similar results that were presented in [2]. We first obtained the results that were presented for the first two scenarios, for the first scenario we obtained a mean final belief of 0.668. When the attacker and the primary user were brought closer together in the second scenario we obtained a mean final belief of 0.89. The differences between the values can be explained by random shadow fading that was used in this technique, the use of

random shadow fading means that the results of two simulations are never going to be the same. The permutation of secondary users is another factor that affects the results, since the secondary users are deployed randomly it is very unlikely that two simulations would feature the same layout. Figure 4.8 shows the results that were obtained by our simulations.

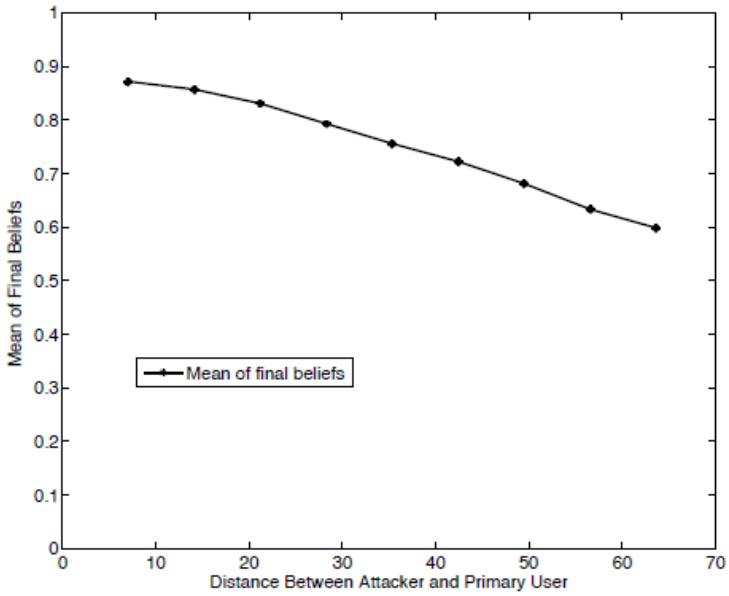


Figure 4.7: Final belief Vs Distance (old technique)

Our results correspond well to the results that were obtained in [2]. To confirm that the simulation results correspond to each other we ran a number of simulations with varying distances between the primary user and the attacker. Figure 4.8 shows the results that were obtained. Once again the results that were obtained are not exactly the same as the results that were obtained in [2]. However, this is to be expected and can again be explained by the random shadowing and the permutation of secondary users.

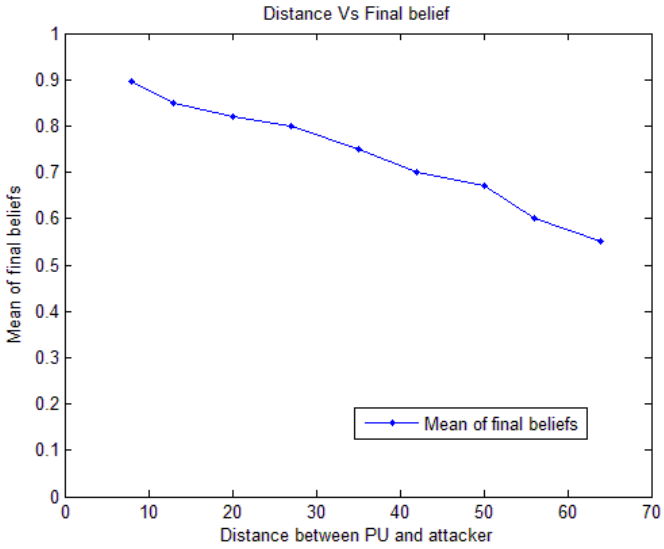


Figure 4.8: Final belief Vs Distance (new technique)

Deficiencies of Existing Technique

The belief propagation algorithm that was proposed in [2] has a number of deficiencies. The key among these is the high computational complexity of the algorithm. The computational complexity corresponds directly to the amount of time it takes for the algorithm to converge. The time it takes the belief propagation algorithm to converge grows exponentially as the number of secondary users in CR network increases. Table one summarises the effects on the computational time as we increase the number of secondary users.

Number of users	Computation time
5	22 seconds
10	101 seconds
15	262 seconds
20	648 seconds
25	1337 seconds
30	2605 seconds

Analysing table 1 it is clear that there is an exponential growth in the computational time of the algorithm as the number of SUs increases. From the table 1 we can conclude that although belief propagation is both reliable and effective in identifying malicious users, it is not feasible for cognitive radio networks with large amounts of secondary users. Figure 4.9 provides a visual representation of the result from table 1.

The primary reason for the high computational complexity of the belief propagation algorithm proposed in [2] is the Kullback Leibler function that is used to evaluate the difference between the primary user probability density function and the attackers probability density function. The KL function evaluates the difference between two function

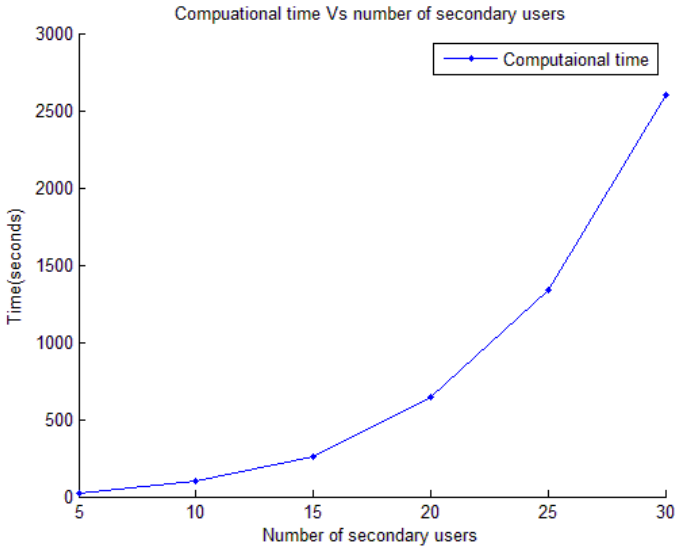


Figure 4.9: Shows the growth in computational time as more SU are added to the network.

using an integral expression. If there are n secondary users in the network the KL function has to be evaluated once for each pair of secondary users, which means that it is calculated n^2 times. Another aspect of the proposed technique that was identified as a possible area of improvement is the compatibility function. To increase the accuracy of the algorithm it is essential that secondary users are able to communicate with as many secondary users as possible. The more secondary users are cooperating the better the accuracy of the final belief is going to be. The compatibility function that is proposed in [2] does not encourage cooperation. The higher the compatibility between pairs of users the more they contribute to each others final beliefs. Given that, the compatibility function is dependent on the distance between secondary users. The closer a pair of secondary users is, the higher their compatibility function will be. In [2] β and C are defined as 2.5, if these values are used then the compatibility function tends to zero too quickly as secondary user distance is increased. The current compatibility function degrades the performance of the technique because it tends close to zero even when the distance between the SUs is as little as 2 meters. If the compatibility function is zero then messages between the two users is zero, which also tends the local function to zero. Which means that SUs are not contributing to each others final belief as much as they should be. We believe that if we want to obtain a greater level of accuracy the proposed compatibility function must be modified.

Chapter 5

A new Belief Propagation based PUEA mitigation algorithm

The primary goal of this project was to develop a new technique that would outperform existing techniques either by increasing the accuracy or reducing of the computational complexity and therefore the computational time. This section presents a new belief propagation mitigation technique that reduces the complexity and increases the accuracy of the belief propagation method presented in [2]. In this chapter, we present changes to the old algorithm and prove that these changes lead to better performance and a significant decrease in the complexity of the old belief propagation algorithm. The most significant improvement is the reduced complexity of the algorithm that is used to calculate the local function for each pair of SUs. With the new algorithm the convergence time for a large number of secondary users is reduced from hours to less than a second. The accuracy of the algorithm is increased by altering the compatibility function. The old compatibility function did not allow an acceptable degree of cooperation between secondary users because its value was very close to zero in most cases. This meant that the messages that were exchanged between users were often meaningless. The new compatibility function insures that there is a greater degree of cooperation between SUs.

5.1 Local Function

The local function that was presented in the old BP technique had a high level of computational complexity and therefore increased the convergence time of the algorithm. The key contribution of the new technique is the development of a simpler more efficient local function. The new local function must enable the BP algorithm to converge faster, while at the same time either increase or maintain the accuracy of the previous algorithm. The following simplified local function decreases the convergence time of the algorithm and provides better results than the previous algorithm:

$$\phi_i = \frac{\mid A - B \mid}{A + B} \tag{5.1}$$

The local function corresponds to the measure of the difference between the RSS measurements from a known primary user against the RSS measurements of a suspect. The closer the correlation between the two measurements the more likely it is that the suspect is a primary user. The old algorithm for doing this was over complicated and used a large integral measurement to compare the simple difference between two RSS measurements. For this reason we developed a new local function which is an approximation of the original function but in a much simpler form. The motivation behind the new local function is to decrease the computational complexity of the algorithm. Instead of performing a complex integral comparison of RSS measurements we use a simple arithmetic expression that provides very similar results with a fraction of the complexity. Using this new local function we are able to achieve a high level of accuracy while significantly reducing the computational complexity. The derivations of A and B (in equations 4.15 and 4.18) remain the same as they were in the original technique.

5.2 Compatibility function

The compatibility function that was presented in the old technique discouraged cooperation between secondary users in the CR network and as a result decreases the accuracy of the final belief. This was primarily due to the fact that the compatibility function returned values that were very close to zero unless secondary users were located very close to each other. As the example, if the distance between the secondary users is as little as 2 meters the old compatibility function returns a value that causes messages that are exchanged between the two users to be meaningless(they tend to 0). After a large number of tests and simulations a reduced version of the compatibility function was derived:

$$\psi_{i,j}(X_i, X_j) = \exp(\frac{d_{X_i,X_j}}{100}) \tag{5.2}$$

This compatibility function insures that secondary users that are close to each other are able to cooperate and share their result effectively to increase the accuracy of the results. The purpose of the new compatibility function is to calibrate beliefs of SUs with different degrees of shadowing. SUs that are distant from each other have different constants, therefore less weight it put on a distant SU in the evaluation of beliefs. This consideration is independent of the statistical distribution of SUs. In belief propagation only one pair of SUs is considered at one time, meaning that the type distribution of SU is not relevant to the proposed compatibility function. The new compatibility function was derived so that there is a larger degree of cooperation between secondary users, this is important because the belief propagation algorithm is based on cooperation. The more cooperation between secondary users the better the algorithm performs.

5.3 Complete algorithm

The algorithm for the new technique is identical to Algorithm 1 presented in the previous section. However, the new algorithm uses the modified equations to calculate the values of the local and compatibility functions. For the local function we use Eq.(5.1) and for the compatibility function we use Eq.(5.2). The result of the belief propagation based strategy depend on the ability of secondary users to communicate with each other. The more cooperation between secondary users exists the more accurate the results the results are going to be.

5.4 Simulations and Results

5.4.1 Computational Complexity / Run time

The most significant improvement offered by the new technique is the reduce complexity and time of convergence. Table 3 summarises the amount of time it takes for the new algorithm to converge against the time it takes for the old algorithm to converge for different numbers of secondary users, the results represent the average times of 10 runs.

Number of users	Comp time Old	Comp Time New
5	22 seconds	0.0491 seconds
10	101 seconds	0.0496 seconds
15	262 seconds	0.0564 seconds
20	648 seconds	0.0682 seconds
25	1337 seconds	0.071 seconds
30	2605 seconds	0.010 seconds

It is evident that the new technique outperforms the old technique by a very large factor. For example, when there are 30 secondary users in the CR network the old technique takes on average 2605 seconds to converge, the new technique takes approximately 0.1 seconds to converge. In this case the new technique reduces the computational time of the algorithm by factor of more than 20 000 times. This shows that the new algorithm offers an enormous reduction in computational complexity and enables the belief algorithm to be used in applications where there is a larger amount of secondary users present in the network. Figure 5.1 presents a visual representation of the results that were obtained by comparing the convergence times of the two algorithms.

In addition to these results, the new technique was tested with much higher numbers of secondary users. The table below represents the results of the additional simulations. It shows that when we add a huge number of secondary users the computational time of the new technique is still low.

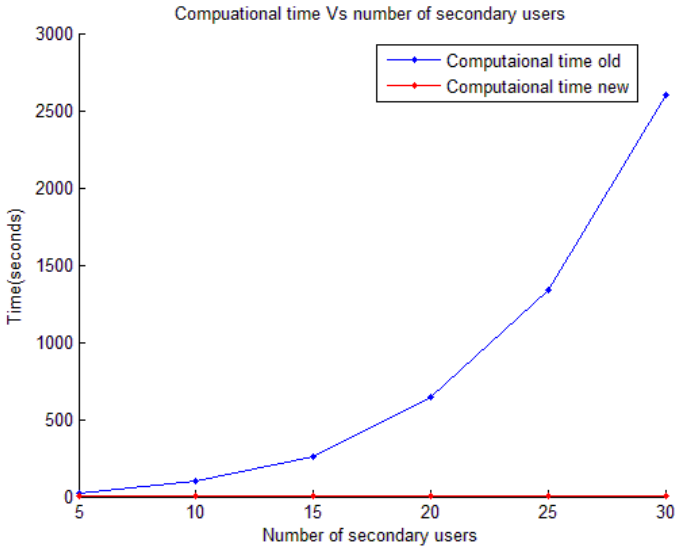


Figure 5.1: Shows the comparison of computational times between the two techniques.

Number of users	Computation time
100	1.4 seconds
300	4.2 seconds
500	11 seconds
1000	30 seconds

5.4.2 Performance

In addition to the reduced computational complexity of the new algorithm, the new algorithm exhibits superior performance to the algorithm presented in [2]. This is primary due to the introduction of a modified compatibility function that allows for a larger degree of cooperation between secondary users. The greater the degree of cooperation between secondary users in the network the lower the chance of false or missed detection of a malicious user. Fig. 3 shows a comparison between the performance of the new algorithm and the performance of the original algorithm.

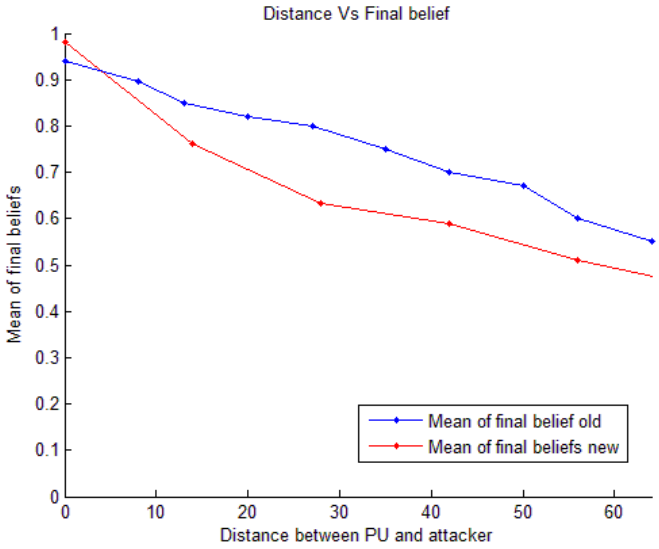


Figure 5.2: Shows the difference between the performance of the old technique and the new technique.

The perfect BP algorithm would result in a final belief value of 1 when the malicious user and the primary user are at the same location and would result in 0 in all other cases. Through analysis of results we observe that the new algorithm has an average final belief that is smaller than the average of the final belief of the old algorithm. This simple and effective comparison shows that the new algorithm is not just less complicated but also detects PUEA with a higher degree of accuracy. We note that a normal distribution was used to model the distribution of secondary users in our simulations.

5.4.3 ROC curves

The received operating characteristics (ROC) curve is used for diagnostic test evaluation. It plots the true positive rate (Sensitivity) against the false positive rate (1-Specificity) for different threshold values. The sensitivity refers to the fraction of primary users that have been correctly identified as primary users. The specificity refers to the fraction of malicious users that have been correctly identified as malicious users. The 1-specificity refers to the fraction of malicious users that are incorrectly identified as primary users [46] [47]. In essence, the ROC curve plots the relationship between the fractions of PU users that are identified correctly as PUs against the fraction of malicious users that are incorrectly identified as PUs for different threshold values. If we set a low threshold we will be able to identify all primary users correctly but will also identify some malicious users incorrectly as primary users, as we increase the threshold the number of malicious users identified as primary users decreases but the number of primary users correctly identified also decreases meaning we would identify a fraction of legitimate primary users as malicious attackers. We use the ROC curve to evaluate the performance of the new technique in different scenarios.

Figure 5.3 shows the ROC curve for the new technique. The curve shows the relationship between the sensitivity and the specificity as the threshold is increased. For this simulation we used 10 secondary users and the distance between the primary user and the attacker is 4 meters. ROC performance is measured according to the area under the curve, the larger the area under the ROC curve the better the algorithm performs, ideally the curve should have an area of 1.

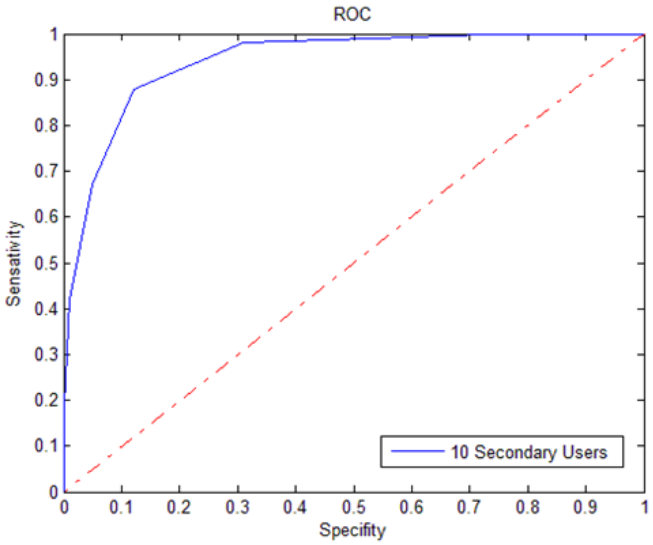


Figure 5.3: ROC curve

The first scenario that was tested is the effect of adding more secondary users to the network. Intuitively, we expect that the performance of the algorithm would increase as the number of SUs increases. The reason for this is that if more users are present they are able to cooperate and obtain more information about the transmitter which would aid in accurate identification of the transmitter as either a malicious user or a PU. Fig 5.4 shows the results that were obtained through simulation.

From the Fig. 5.4 it is evident that as the number of secondary users increases the area under the curve also increases which means that the overall performance of the algorithm also increases. The distance between the malicious attacker and the PU is critical, the closer the attacker is to the PU the harder it is to distinguish the attacker from the PU. Fig 5.5 demonstrates the performance of the algorithm as the distance between the attacker and the malicious user is decreased. Intuitively, we expect that as the attacker moves closer to the PU we would get worse results. From our results we see that when the attacker is 4 meters away from the PU we get an area under the curve that is very close to one which corresponds to very good results, as we move the PU and the attacker closer together we see that the results begin to deteriorate. The reason for this is that as the at-

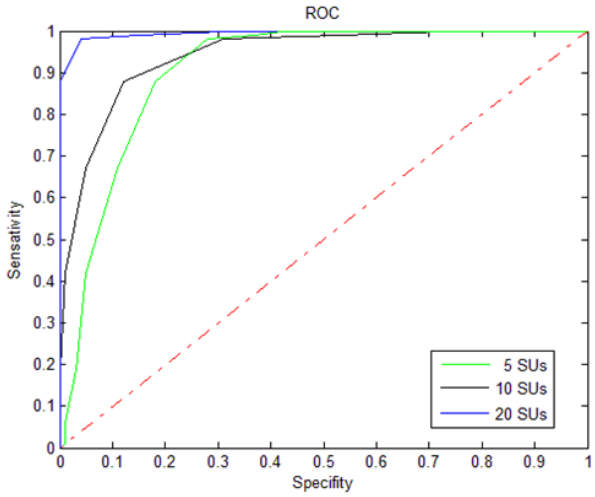


Figure 5.4: ROC curve corresponding to different numbers of SUs.

tacker moves closer to the PU it is harder to distinguish between the PU and the attacker.

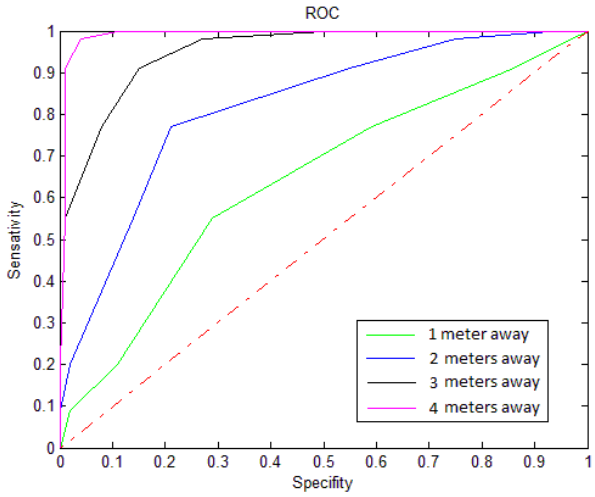


Figure 5.5: ROC curve corresponding to different distances between PU and attacker.

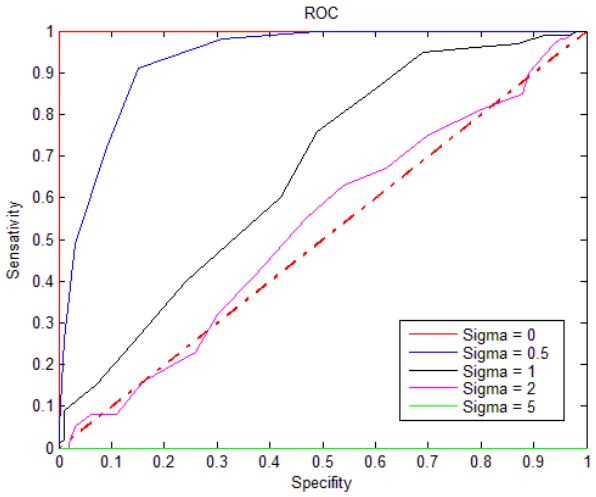


Figure 5.6: ROC curve corresponding to varied shadowing.

Figure 5.6 demonstrates the effects of shadowing on the new algorithm. From Fig. 5.6 we see that when there is no shadowing the algorithm is perfect and has an area under the graph of 1. As we increase the variance we see that the performance of the algorithm degrades. When the variation is increased to 5 we see that the performance of the algorithm drops significantly.

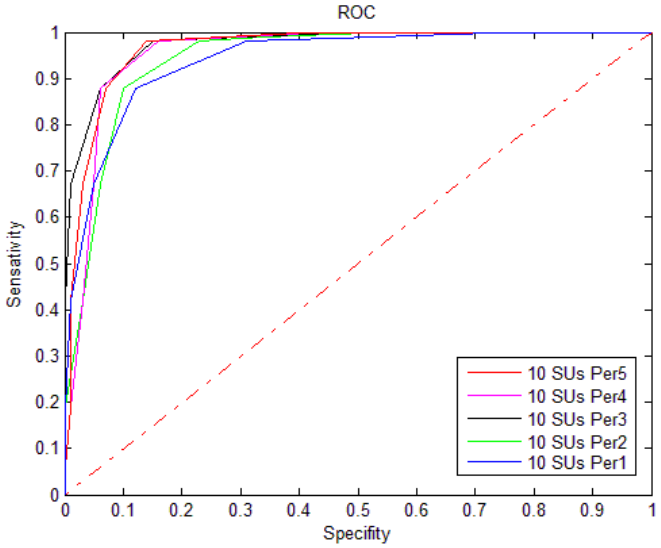


Figure 5.7: ROC curve corresponding to different permutations of SUs.

Since the location of secondary users is thought to be random. Different permutations of secondary user locations will yield different results. However, it is important that the difference in the results of the permutation is as small as possible. Fig. 5.7 shows the difference in performance of 5 random permutations of a network with 10 secondary users.

From the results it is evident that the results are different for each permutation, but that the results still correspond well with each other. We see that the difference in the area under the curve varies by less than 10 per cent.

Throughout this chapter we have presented results that show that the new belief propagation technique is a feasible solution to the PUEA problem. We have shown that the new technique performs with a higher degree of accuracy and a lower degree of complexity than the existing mitigation technique. We have shown that it is scalable and offers high efficiency even when a large number of secondary users are present on the network. The ROC plots that are presented in this chapter give insight into how the algorithm operates under changing conditions. We show that the new algorithm operates well in a shadowing environment, with varied SU locations and when the PU and the attacker are close to each other. The simulation results presented show that the new technique is a feasible solution for the mitigation of primary user emulation attacks against cognitive radio networks.

Chapter 6

Conclusion

In this paper we present a belief propagation based algorithm to combat the effects of primary user emulation attacks on cognitive radio networks. We introduce key improvements to the algorithm described in [2] in relation to both performance and computational complexity. Through simulation we were able to show that our technique has lower complexity and improved accuracy relative to the technique in [2]. We have shown that the new technique reduces the time of convergence of the BP algorithm from hours to less than a few seconds. Furthermore, despite the simplification of the algorithm we were able to accurately distinguish between primary user and primary user emulation transmissions. These improvements are a direct result of the new local and compatibility functions, which reduce complexity and allow a greater degree of cooperation between secondary users on the CR network. The new algorithm is scalable, efficient, and effective and may be implemented in a low complexity secondary user terminal. The new algorithm provides a significant step forward in the mitigation of primary user emulation attacks in cognitive radio networks using belief propagation. From the results presented in this paper we conclude that our new technique presents a significant improvement over the old technique presented in [2]. The new technique offers a higher degree of accuracy and at the same time operates with a lower degree of complexity making it a clear improvement on the old technique. The new technique does not require expensive hardware or software implementation to provide accurate results.

Appendix A

Triangulation Method

```
%Points on the plain
%PU = [472 371];

PU = [100 600];
Attacker = [472 371]; % original values 472 and 371
SUs = [400 300; 600 300; 350 480; 650 450];
num = 3;%number of secondary nodes
distanceSU = zeros(1,num); % an array for the distance values of the secondary
%users
distanceA = zeros(1,num+1);
%%pathloss = zeros(1,num); % not needed for now we assume that the passloss
%is 2 for now
RSS = zeros(1,num+1);
radius = zeros(1,num);
center = zeros(num,2);
frequency = 500; % frequency in MHz
pathloss = 2;
TransmitPower = 30; % 1 watt or 30 dBm
s = 0; % shadowing for this example is 0
n = zeros(1,num);
IntersectionPoint = zeros(1,2);



% distance between secondary user nodes
for i=1:num;

    d = [SUs(1,1), SUs(1,2);SUs(i+1,1), SUs(i+1,2)]; % compares distance
    %% between SU1 and every other node
    dis = pdist(d); %calculates distance between node 1 and other nodes
```

```
    distanceSU(i) =dis; %updates distance

end

%distance between the attacker and each SU node
for i=1:num+1;


    d = [Attacker(1,1), Attacker(1,2);SUs(i,1), SUs(i,2)];
    dis = pdist(d);
    distanceA(i) = dis; %Distance between the attacker and each node
    %pathloss(i) = log10(distanceA(i)) + log10(frequency) - 27.55;% simple free
    %space path loss expressed in dB for meters and MHz for now pathloss is 2
    %pathloss(i) = pathloss(i) +30; % convert from dbm to db
    RSS(i)= TransmitPower - pathloss * log10(distanceA(i)) - s; % RSS values for
    %each node in dBm
end

% circle parameters the centre point and the radius of the circle
for i=1:num;

    n(i) = 10^((RSS(1)-RSS(i+1))/pathloss); % difference in RSS between two sus
    radius(i) = (n(i)*distanceSU(i))/(((n(i))^2)-1);%radius
    center(i,1)= (((n(i))^2)* SUs(1,1) - SUs(i+1,1))/(((n(i))^2)-1);
    %x coordinate
    center(i,2)= (((n(i))^2)* SUs(1,2) - SUs(i+1,2))/(((n(i))^2)-1);
    %y coordinate

end

for i=1:num;
viscircles([center(i,1),center(i,2)],radius(i)); % creates actual circle
end

%Find the intersect points of circle 1 and 2, and of circle 1 and 3
for i=1:1;
[xout,yout] = circcirc(center(i,1),center(i,2),radius(i),center(i+1,1),...
center(i+1,2),radius(i+1));
[xout1,yout1] = circcirc(center(i+1,1),center(i+1,2),radius(i+1),...
center(i+2,1),center(i+2,2),radius(i+2));
end
%round the results off to nearest number
xout = round(xout);
```

```
xout1 = round(xout1);
yout = round(yout);
yout1 = round(yout1);

%check which points correspond to the actual intesect points of the three
%circles
for i=1:1;

    if ((xout(1)== xout1(2))) && (yout(1) == yout1(2))||((xout(1)== xout1(1)) &&
    (yout(1) == yout1(1)))

        intersectionPoint(1,1)= xout(1);
        intersectionPoint(1,2)= yout(1);

    elseif (xout(2)== xout1(1) && yout(2) == yout1(1)) || (xout(2)== xout1(2) &&.
     yout(2) == yout1(2));

        intersectionPoint(1,1)= xout(2);
        intersectionPoint(1,2)= yout(2);
    end

end

%plotting PU
hold on;
plot(PU(:,1),PU(:,2),'LineStyle','none', 'Marker','.', 'Color','Blue');
text(PU(:,1), PU(:,2), 'PU1', 'HorizontalAlignment','left', 'VerticalAlignment'..
,'bottom','Color','blue') % prints and aligns labels
%plot attacker
plot(Attacker(:,1),Attacker(:,2),'LineStyle','none', 'Marker','.', 'Color','red')
text(Attacker(:,1), Attacker(:,2), 'Attacker', 'HorizontalAlignment','left'...
, 'VerticalAlignment','bottom','Color','red') % prints and aligns labels
%plotting SUs
plot(SUs(:,1),SUs(:,2),'LineStyle','none', 'Marker','.', 'Color','black');
% plots each point on the plain
str = num2str((1:num+1)','SU%d');    %defines labels for each point
text(SUs(:,1), SUs(:,2), str, 'HorizontalAlignment','left', 'VerticalAlignment'..
,'bottom','Color','black ') % prints and aligns labels
axis([0 700 0 700]);
```

Appendix B

Original Belief Propagation code

```
tic
%Points on the plain
PU = [5 5]; % they assume there is only one primary user
%rng(0,'twister');
Attacker = [50 50];%(100).*rand(1,2);
%SUs = [81 63; 90 9; 12 27; 91 54];
SUs = (100).*rand(10,2);% generate 30 random SUs in the 100x100 area
sizeofattacker = size(Attacker);
numAtt = sizeofattacker(1,1);
%SUs = [81 63; 90 9; 12 27; 91 54];
sizeofSU = size(SUs); % calculates the number of SUs
numSU = sizeofSU(1,1);%number of secondary nodes
sizeofPU = size(PU);
numPU = sizeofPU(1,1);
distanceA = zeros(numAtt,numSU); % Distance betweent each SU and the attacker
distancePU = zeros(numPU,numSU); % Sistance between each SU and PU
%%pathloss = zeros(1,num-1); % not needed for now we assume that the passloss is
%2 for now
RSSattacker = zeros(1,numSU);
frequency = 500; % frequency in MHz
pathloss = 2.5; % given passloss constant
TransmitPower = 0.1; % 0.1 watt or 20 dBm
n = zeros(1,numSU-1);
distanceSU = zeros(numSU,numSU);
syms x;
%q = zeros(numSU,numSU);
PDFpu = zeros(numSU,numSU);
PDFA = zeros(numSU,numSU);
KLint = zeros(numSU,numSU);
LocFun = zeros(numSU,numSU);
```

```
CompFun= zeros(numSU,numSU);
JointProb= zeros(numSU,numSU);
msg = zeros(numSU,numSU);
msgtotal = zeros(numSU,numSU);
belief = zeros(1,numSU);
B = zeros(numSU,numSU);
A = zeros(numSU,numSU);
C = zeros(numSU,numSU);
h = zeros(numPU,numSU); %shadowing variable from SU to PU
numberIte = 3;
msg123 = zeros(numSU,numSU);
msgsum = zeros(1,numSU);
msgprod = zeros(1,numSU);

% shadowing variable
for i=1:numPU;
    for j=1:numSU;
       b = normrnd(0,0.5); %normal distribution mean 0 variance 1
       a = (log(10))/10; % definign a as a constant
       h(i,j) = exp(a*b); %A 2d random variable with mean 0 and variance 1
    end
end
% RSS and distance between the attacker and each SU node
for i=1:numAtt;

     for k=1:numSU;
        d = [Attacker(i,1), Attacker(i,2);SUs(k,1), SUs(k,2)];
        dis = pdist(d);
        distanceA(i,k) = dis; %Distance between the attacker and each node
        %pathloss(i) = log10(distanceA(i)) + log10(frequency) - 27.55;% simple
        %free space path loss expressed in dB for meters and MHz for now
        %pathloss is 2
        %pathloss(i) = pathloss(i) +30; % convert from dbm to db
        %RSSattacker(i,k)= TransmitPower - pathloss *
        %log10(distanceA(i,k))-h(i,k); % RSS values for each node in dBm
        RSSattacker(i,k) = TransmitPower*(distanceA(i,k)^(-pathloss))*h(i,k);
     end
end

% Belief Propagation

%Distances between each SU and the PU.
for i=1:numPU
```

```
    for k=1:numSU;
        d = [PU(i,1),PU(i,2);SUs(k,1),SUs(k,2)];
        dis = pdist(d);
        distancePU(i,k) = dis;
    end
end

% distances between SUs
for i=1:numSU
        for k =1:numSU;
        d = [SUs(i,1), SUs(i,2);SUs(k,1), SUs(k,2)];
        dis = pdist(d); %calculates distance between a given node and all
        %others on the network
        distanceSU(i,k) = dis; %updates matrix with distance
        end

end

% %PDF Primary user
for i=1:numPU;
    for j=1:numSU;
        for k =1:numSU;
            B(j,k) = (distancePU(i,j)/distancePU(i,k))^(-pathloss);
            %calculates values of B (Distance between SUs and PU)
            A(j,k) = (distanceA(i,j)/distanceA(i,k))^(-pathloss);
            %A(j,k) = (RSSattacker(i,j)/RSSattacker(i,k))/(pi/2);
            %calculate values of A using eq22
        end
    end
end


for i=1:numSU;
    for j=1:numSU;
       for p=1:2; %distance metric from Sam
       KLint(i,j) = (abs((A(i,j)^p)-(B(i,j)^p)))^(1/p);
       end
    end
end

%local function
for i=1:numSU;
```

```
    for j=1:numSU;
        if(i~=j);
           LocFun(i,j) = exp(-KLint(i,j)); % local function for each pair
           %of SU nodes eq 29
        else
            LocFun(i,j) = 0;
        end

    end
end


% Compatibility function
for i=1:numSU;
    for j=1:numSU;
        if(i ~= j);
                CompFun(i,j) = exp(-0.01*(distanceSU(i,j)));
        end
    end
end

%Generate messages
for k=1:2;

    for i=1:numSU;
        msginit = 1;%init msg
        initbel = 1; %intial belief
        for j=1:numSU;
            if(k == 1);%first message where the previous msg is 1
                if(i ~= j);
                    if(distanceSU(i,j) < 20);%checks if the users are within
                    %the 20 meter range
                        msg(i,j)= CompFun(i,j)*LocFun(i,j); % product of the
                        % comp and loc function
                        C = 1/((2*CompFun(i,j))*msginit);
                        msgtotal(i,j) = msg(i,j)*msginit;%update msg
                    else %if SUs are not within the 20 meter range msg between
                    % them is 0
                        msg(i,j) = CompFun(i,j)*LocFun(i,j);
                        msgtotal(i,j) = 0;
                    end

                end
```

```
            else % iteration msgs
                if(i~=j);
                    if(distanceSU(i,j) < 20);
                        C = 1/(2*CompFun(i,j)*msgtotal(i,j));
                        msgtotal(i,j) =  C * msg(i,j)*msgtotal(i,j);
                    else
                        msgtotal(i,j) = 0;
                    end
                end
            end
        end
        %norm constant C
%         normmsg = sum(msgtotal,2);%update the sum of all msgs at each node
%       %after each iteration for new C = 1/msgtotal
%         for a=1:numSU;
%             msgtotal(i,a) = msgtotal(i,a) /normmsg(i); %implementing c
        %and normalising the msgs at each node
%         end
        %beliefs
        temp = msgtotal;
        temp(temp == 0) = 1; %insures that the product does not equal to zero
        temp1 = LocFun;
        temp1(temp1==0) = 10; %insure that the local min is not zero
        msgprod = prod(temp,2);%calculates the product of the msgs going
        %into each SU
        localmax = max(LocFun,[],2); %calculates the local function
        %for the two
        %closest SUs
        localmin = min(temp1,[],2);%calculates the local function minimum

        for a=1:numSU;
           msgsum = sum(msgtotal,2);
           if (msgsum(a) == 0);
                belief(a) = localmax(a);
           else
               K = 1/msgprod(a);
               belief(a) =  K*localmax(a) * msgprod(a);
           end
        end
    end
    belief
    fb = (sum(belief)/numSU)
end
```

```
% %Calc Belief
% for i=1:numSU;
%     temp = LocFun; %using temp to cancel out the zero terms while
        %insuring the locfun is unchanged
%     temp(temp==0) = 1000; %changes all zeo terms in locfun to 1000
        %to insure they are not confused as the min
%     localmin =  min(temp,[],2); % looking for the minimum local
        % function for the final belief EQ19
%     %add all msgs for each node
%     s = sum(msgtotal,2);
%     msgsum(i) = s(i);
%     k = 1/msgsum(i); % norm constant
%     belief(i) = k*localmin(i)*msgsum(i); %final belief
%
% end
```

```
%plotting PU
grid on;
hold on;
plot(PU(:,1),PU(:,2),'LineStyle','none', 'Marker','.', 'Color','Blue');
text(PU(:,1), PU(:,2), 'PU1', 'HorizontalAlignment','left', 'VerticalAlignment'..
,'bottom','Color','blue') % prints and aligns labels
%plot attacker
plot(Attacker(:,1),Attacker(:,2),'LineStyle','none', 'Marker','.', 'Color'...
,'red');
text(Attacker(:,1), Attacker(:,2), 'Attacker', 'HorizontalAlignment','left'...
, 'VerticalAlignment','bottom','Color','red') % prints and aligns labels
%plotting SUs
plot(SUs(:,1),SUs(:,2),'LineStyle','none', 'Marker','.', 'Color','black');
% plots each point on the plain
str = num2str((1:numSU)','SU%d');    %defines labels for each point
text(SUs(:,1), SUs(:,2), str, 'HorizontalAlignment','left'...
, 'VerticalAlignment','bottom','Color','black ') % prints and aligns labels
axis([0 100 0 100]);
```

```
toc
```

Appendix C

New Belief Propagation based code

```
tic
%Points on the plain
PU = [40 40]; % they assume there is only one primary user
%rng(0,'twister');
Attacker = [50 50];%(100).*rand(1,2);
%SUs = [81 15; 90 97; 12 95; 91 48;63 80;9 14; 27 42; 54 91; 95 79; 96 95];
SUs = (100).*rand(10,2);% generate 30 random SUs in the 100x100 area
sizeofattacker = size(Attacker);
numAtt = sizeofattacker(1,1);
sizeofSU = size(SUs); % calculates the number of SUs
numSU = sizeofSU(1,1);%number of secondary nodes
sizeofPU = size(PU);
numPU = sizeofPU(1,1);
distanceA = zeros(numAtt,numSU); % Distance betweent each SU and the attacker
distancePU = zeros(numPU,numSU); % Sistance between each SU and PU
%%pathloss = zeros(1,num-1); % not needed for now we assume that the passloss
%is 2 for now
RSSattacker = zeros(1,numSU);
RSSpu = zeros(1,numSU);
frequency = 500; % frequency in MHz
pathloss = 2.5; % given passloss constant
TransmitPower = 0.1; % 0.1 watt or 20 dBm
n = zeros(1,numSU-1);
distanceSU = zeros(numSU,numSU);
syms x;
%q = zeros(numSU,numSU);
% PDFpu = zeros(numSU,numSU);
% PDFA = zeros(numSU,numSU);
KLint = zeros(numSU,numSU);
LocFun = zeros(1,numSU);
```

```
CompFun= zeros(numSU,numSU);
JointProb= zeros(numSU,numSU);
msg = zeros(numSU,numSU);
msgtotal = zeros(numSU,numSU);
belief = zeros(1,numSU);
B = zeros(1,numSU);
A = zeros(1,numSU);
C = zeros(numSU,numSU);
h = zeros(numPU,numSU); %shadowing variable from SU to PU
hp = zeros(numPU,numSU);
numberIte = 3;
msg123 = zeros(numSU,numSU);
msgsum = zeros(1,numSU);
msgprod = zeros(1,numSU);
abc = zeros(1,numSU);

% shadowing variable attacker
for i=1:numPU;
    for j=1:numSU;
       b = normrnd(0,0.5); %normal distribution mean 0 variance 1
       a = (log(10))/10; % definign a as a constant
       h(i,j) = exp(a*b); %A 2d random variable with mean 0 and variance 1
    end
end
% shadowing variable PU
for i=1:numPU;
    for j=1:numSU;
       c = normrnd(0,0.5); %normal distribution mean 0 variance 1
       d = (log(10))/10; % definign a as a constant
       hp(i,j) = exp(c*d); %A 2d random variable with mean 0 and variance 1
    end
end
% RSS and distance between the attacker and each SU node
for i=1:numAtt;

     for k=1:numSU;
        d = [Attacker(i,1), Attacker(i,2);SUs(k,1), SUs(k,2)];
        dis = pdist(d);
        distanceA(i,k) = dis; %Distance between the attacker and each node
        %pathloss(i) = log10(distanceA(i)) + log10(frequency) - 27.55;%
        %simple free space path loss expressed in dB for meters and MHz for
        %now pathloss is 2
        %pathloss(i) = pathloss(i) +30; % convert from dbm to db
```

```
        %RSSattacker(i,k)= TransmitPower - pathloss * log10(distanceA(i,k))...
        -h(i,k); % RSS values for each node in dBm
        RSSattacker(i,k) = TransmitPower*(distanceA(i,k)^(-pathloss))*h(i,k);
     end
end

%Distances between each SU and the PU.
for i=1:numPU

    for k=1:numSU;
        d = [PU(i,1),PU(i,2);SUs(k,1),SUs(k,2)];
        dis = pdist(d);
        distancePU(i,k) = dis;
        RSSpu(i,k) = TransmitPower*(distancePU(i,k)^(-pathloss))*hp(i,k);
    end
end

% distances between SUs
for i=1:numSU
        for k =1:numSU;
        d = [SUs(i,1), SUs(i,2);SUs(k,1), SUs(k,2)];
        dis = pdist(d); %calculates distance between a given
        % node and all others on hte network
        distanceSU(i,k) = dis; %updates matrix with distance
        end

end

% %PDF Primary user
for i=1:numPU;
    for j=1:numSU;
%             B(j) = (distancePU(i,j))^(-pathloss); %calculates values
            %of B (Distance between SUs and PU)
%             A(j) = (distanceA(i,j))^(-pathloss);
            A(j) = (RSSattacker(i,j))/(pi/2);%calculate values of A using eq22
            B(j) = (RSSpu(i,j))/(pi/2);
    end
end

%Local Function and new algorithm
for i=1:numSU;
    abc(i) = (abs(B(i)-A(i)))/(B(i)+A(i));
    LocFun(i) = exp(-abc(i)); % local function for each pair of SU nodes eq 29
```

```
end

% Compatibility function
for i=1:numSU;
    for j=1:numSU;
        if(i ~= j);
                CompFun(i,j) = exp(-(distanceSU(i,j)/100));
        end
    end
end

%Generate all the messages
for i=1;numSU;
    for j=1:numSU;
        if(distanceSU(i,j) < 20);
            msg(i,j) = LocFun(j)*CompFun(i,j);
        else
            msg(i,j) = 0;
        end
    end
end

for i=1:numSU;
    msgsum = sum(msg,2); %we check to see if a SU has any msgs from other SUs
    if (msgsum(i) == 0);
        belief(i) = LocFun(i);
    else
        temp = msg;
        temp(temp == 0) =1;
        msgproduct = prod(temp,2);
        for j=1:numSU;
            if(msg(i,j) > 0);
                belief(i) = (LocFun(i) + msgproduct(i))/2;
            end
        end
    end
end

belief;
fb = (sum(belief)/numSU)
```

```
%plotting PU
grid on;
hold on;
plot(PU(:,1),PU(:,2),'LineStyle','none', 'Marker','.', 'Color','Blue');
text(PU(:,1), PU(:,2), 'PU1', 'HorizontalAlignment','left'...
, 'VerticalAlignment','bottom','Color','blue') % prints and aligns labels
%plot attacker
plot(Attacker(:,1),Attacker(:,2),'LineStyle','none', 'Marker','.', 'Color','red')
text(Attacker(:,1), Attacker(:,2), 'Attacker', 'HorizontalAlignment','left'...
, 'VerticalAlignment','bottom','Color','red') % prints and aligns labels
%plotting SUs
plot(SUs(:,1),SUs(:,2),'LineStyle','none', 'Marker','.', 'Color','black');
% plots each point on the plain
str = num2str((1:numSU)','SU%d');    %defines labels for each point
text(SUs(:,1), SUs(:,2), str, 'HorizontalAlignment','left'...
, 'VerticalAlignment','bottom','Color','black ') % prints and aligns labels
axis([0 100 0 100]);


toc
```

Chapter 7

Acronyms and Abbreviations

CR	Cognitive Radio
PUEA	Primary User Emulation Attack
MAC	Media Access Control
SSDFA	Spectrum Sensing Data Falsification Attacks
RSS	Received Signal Strenght
MATLAB	Matrix Laboratory
TDOA	Time difference of arrival
FDOA	Frequency difference of arrival
BS	Base Station
SU	Secondary User
PU	Primary User
ITU	International Telecommunication Union
SDR	Software Defined Radio
MIMO	Multiple-Input and Multiple-Output
AWGN	Additive White Gaussian Noise
FPGA	Field-Programmable Gate Array
DSP	Digital Signal Processors
GPP	General Purpose Processors
SoC	System on Chip
CN	Cognitive Network
CRN	Cognitive Radio Network
DoS	Denial of Service
TCP	Transmission Control Protocol
BP	Belief Propagation
MRF	Markov Random Field
PDF	Probability Density Function
ROC	Received Operating Characteristics
FCC	Federal Communications Commission

Bibliography

[1] S. Arkoulis L. Kazatzopoulos C. Delakouridis G.F. Marias, "Simulation framework for security threats in cognitive radio networks," *Communications, IET (Volume:6 , Issue:8), pages 984 to 990*, July, 2011.

[2] Zhou Yuan, Dusit Niyato, Husheng Li, Ju Bin Song and Zhu Han, "Defeating Primary User Emulation Attacks Using Belief Propagation in Cognitive Radio Networks," *Selected Areas in Communications, IEEE Journal (Volume:30 , Issue:10), pages 1850 to 1860*, November, 2012.

[3] E. Hossain, D. Niyato and Z. Han, "*Dynamic Spectrum Access in Cognitive Radio Networks*," June, 2009.

[4] J. Mitola III and G. Q. Maguire, Jr., "Cognitive radio: Making software radios more personal," *IEEE Personal Communications, vol. 6, pp. 13 to 18*, August, 1999.

[5] Wednel Cadeau and Xiaohua Li, "Jamming probabilities and throughput of cognitive radio communications against a wideband jammer," *Information Sciences and Systems (CISS), pages 1 to 6*, March, 2013.

[6] Husheng Li and Zhu Han , "Dogfight in Spectrum: Jamming and Anti-Jamming in Multichannel Cognitive Radio Systems," *Global Telecommunications Conference, GLOBECOM, pages 1 to 6*, December, 2009.

[7] Badr Benmammar, Asma Amraoui and Francine Krief, "A Survey on Dynamic Spectrum Access Techniques in Cognitive Radio Networks," *International Journal of Communication Networks and Information Security (IJCNIS) Vol. 5, No. 2, pages 68 to 79*, August, 2013.

[8] Tevfik Yucek and Huseyin Arsla, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications ," *Communications Surveys & Tutorials, IEEE (Volume:11 , Issue:1), pages 116 to 130*, March, 2009.

[9] Manman Dang, Zhifeng Zhao and Honggang Zhangy, "Detection of Primary User Emulation Attacks Based on Compressive Sensing in Cognitive Radio Networks ," *Wireless Communications & Signal Processing (WCSP), pages 1 to 5*, October, 2013.

[10] Ruiliang Chen, Jung-Min Park and Jeffrey H. Reed, "Defense against Primary User Emulation Attacks in Cognitive Radio Networks," *Selected Areas in Communications, IEEE Journal on (Volume:26 , Issue:1) , pages 25 to 37*, January 2008.

[11] Mathworks, "MATLAB the Language of Technical Computing."

[12] Ruiliang Chen and Jung-Min Park, "Ensuring Trustworthy Spectrum Sensing in Cognitive Radio Networks, Networking Technologies for Software Defined Radio Networks," *Networking Technologies for Software Defined Radio Networks, pages 110 to 119*, September, 2006.

[13] Meena Thanu, "Detection of Primary User Emulation Attacks in Cognitive Radio Networks," *Collaboration Technologies and Systems (CTS), pages 605 to 608*, May, 2012.

[14] Lianfen Huang, Liang Xie, Han Yu, Wumei Wang and Yan Yao, "Anti-PUE Attack Based on Joint Position Verification in Cognitive Radio Networks," *Communications and Mobile Computing (CMC), pages 169 to 173*, April, 2010.

[15] Xiao Zhou, Yang Xiao and Yuanyuan Li, "Encryption and Displacement Based Scheme of Defense against Primary User Emulation Attack," *Wireless, Mobile & Multimedia Networks (ICWMMN 2011), 4th IET, pages 44 to 49.*

[16] Federal Communications Commission, "FCC. Et docket no.02-135. Spectrum policy task force (SPTF) report. Tech. rep," February, 2002.

[17] International Telecommunications union Report ITU-R SM.2152, "Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS), page 3 ," September, 2009.

[18] Khaled Ben Letaief and Wei Zhang, "Cooperative Communications for Cognitive Radio Networks," *Proceedings of the IEEE (Volume:97 , Issue:5, pages 878 to 893)*, May, 2009.

[19] Nemanja Vucevic, Ian F. Akyildiz and Jordi Perez-Romero, "Dynamic Cooperation selection in cognitive radio networks," *Ad Hoc Networks Volume 10, Issue 5, pages 789 to 802*, February, 2011.

[20] Kevin Chan, "Spectrum sensing, detection and optimization in cognitive radio for non-stationary primary user signals, *PDH thesis*," September, 2012.

[21] Bruce Fette, "Cognitive Radio Technology, 2nd Edition," March, 2007.

[22] Hseyin Arrssllaann , "Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems EmphChapter 4, pages 82 to 112," June, 2007.

[23] Khattab A. Perkins D. Bayoumi M., "Cognitive Radio Networks from Theory to Practice *Chapter 2, pages 11 to 32*," August, 2012.

[24] P.Kolodzy, "Proceedings of the Defense Advanced Research Projects Agency," 2001.

[25] S.Gaur, J.S. Jiang, M. Ingram, M.Demirkol , "Interfering MIMO links with stream control and optimal antenna selection," *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE (Volume:5), pages 3138 to 3142*, December, 2004.

[26] Rehan Ahmed and Yasir Arfat Ghous, "Detection of vacant frequency bands in cogntive radio, *Blekinge Institute of Technology*," May, 2010.

[27] Marija Matinmikko, Marko Hyhty, Miia Mustonen, Heli Sarvanko, Atso Hekkala, Marcos Katz, Aarne Mmmel, Markku Kiviranta, Aino Kautio, "Cognitive radio: An intelligent wireless communication system, *Research Report*," March, 2008.

[28] I.F. Akyildiz, B.F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication, vol. 4 no. 1 pp. 40 to 62*, November, 2011.

[29] A. Garhwal, and P. P. Bhattacharya, "A Survey on Dynamic Spectrum Access Techniques for Cognitive Radio," *International Journal of Next-Generation Networks, vol. 3, no. 4, pp. 15 to 32,*, September, 2012.

[30] S. Ziafat, W. Ejaz, and H. Jamal, "Spectrum sensing techniques for cognitive radio networks: Performance analysis," *IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals, pages 1 to 4*, August, 2011.

[31] S.Shobana, R.Saravanan and R.Muthaiah, "Matched Filter Based Spectrum Sensing on Cognitive Radio for OFDM WLANs," *International Journal of Engineering and Technology (IJET), pages 142 to 146*, February, 2013.

[32] Pradeep Kumar Verma, Sachin Taluja and Rajeshwar Lal Dua, "Performance analysis of Energy detection, Matched filter detection and Cyclostationary feature detection Spectrum Sensing Techniques," *International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5, pages 1296 to 1301*, September, 2013.

[33] D. abri, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios,," *Proceedings of Asilomar Conference 2004, pp. 772 to 776*, November, 2004.

[34] Tevfik Yucek and Huseyin Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications," *Communications Surveys amd Tutorials, IEEE. Volume : 11, Issue: 1, pages 116 to 130* , March, 2014.

[35] Wireless-innovation, "Wireless innovation, software defined radio SDR[online]," 2012.

[36] Maksym Girnyk, "Cooperative Communication for Multi-User Cognitive Radio Networks," *Thesis in Telecommunications Stockholm, Sweden*, June, 2012.

[37] Lorenza Giupponi and Christian Ibars, "Cooperative Cognitive Systems, page 340," November, 2009.

[38] Bing Xia, Muhammad Husni Wahab, Yang Yang Zhong Fan and Mahesh Sooriyabandara, "Cognitive Spectrum and its Security Issues," *Next Generation Mobile Applications, Services and Technologies, pages 565 to 570* , September, 2014.

[39] Jack L. Burbank, "Security in Cognitive Radio Networks: The Required Evolution in Approaches to Wireless Network Security," *CrownCom 2008*, May, 2008.

[40] Alexandros G. Fragkiadakis, Elias Z. Tragos and Ioannis G. Askoxylakis, "A Survey on Security Threats and Detection Techniques in Cognitive Radio Networks," *Communications Surveys & Tutorials, IEEE (Volume:15 , Issue:1), pages 428 to 445*, January, 2013.

[41] Wassim El-Hajj, Haidar Safa and Mohsen Guizani, "Survey of Security Issues in Cognitive Radio Networks," *Electronics and Communication Systems (ICECS), pages 1 to 5*, February, 2014.

[42] Juan Hernandez-Serrano, Olga Len and Miguel Soriano, "Modeling the Lion Attack in Cognitive Radio Networks," *EURASIP Journal on Wireless Communications and Networking, Vol.2011, Article ID 242304, 10 pages*, August, 2011.

[43] A. Pandharipande, "Wireless RANs: Technology Proposal Package for IEEE 802.22, IEEE 802.22 WG on WRANs," November, 2005.

[44] Ruiliang Chen, Jung-Min Park, Y. Thomas Hou and Jeffrey H. Reed, "Toward Secure Distributed Spectrum Sensing in Cognitive Radio Networks," *IEEE Communications Magazine, Vol.46, No.4*, April, 2008.

[45] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding Belief Propagation and its Generalizations, Exploring Artificial Intelligence in the New Millennium, Chap. 8, pp. 2282 to 2312, Science and Technology Books," *Exploring Artificial Intelligence in the New Millennium, Chap. 8, pp. 2282 to 2312*, 2003.

[46] ROC Curves, "ROC curve analysis in MedCalc[online], http://www.medcalc.org/manual/roc-curves.php,"

[47] ROC Curves, "The Area under an ROC Curve[online], http://gim.unmc.edu/dxtests/roc3.htm,"