Hearing is Believing: Detecting Mobile Primary User Emulation Attack in White Space

Shaxun Chen, Kai Zeng, Prasant Mohapatra
Department of Computer Science, University of California, Davis, CA 95616
{sxch, kaizeng, pmohapatra}@ucdavis.edu

Abstract— **In cognitive radio networks, an adversary transmits signals whose characteristics emulate those of primary users, in order to prevent secondary users from transmitting. Such an attack is called primary user emulation (PUE) attack. There are two main types of primary users in white space: TV towers and wireless microphones. Existing work on PUE attack detection focused on the first category. However, for the latter category, primary users are mobile and their transmission power is low. These unique properties of wireless microphones introduce great challenges and existing methods are not applicable. In this paper, we propose a novel method to detect the PUE attack of mobile primary users. We exploit the correlations between RF signals and acoustic information to verify the existence of wireless microphones. The effectiveness of our approach is validated through extensive real-world experiments. It shows that our method achieves both false positive rate and false negative rate lower than 0.1.**

I. Introduction

The popularity of wireless communication and ever-increasing wireless traffic have put significant pressure on spectrum utilization. Recognizing the significance of spectrum shortage, the Federal Communications Commission (FCC) released analogue TV bands, often referred to as white space, to unlicensed users on a non-interference basis.

To access white space, unlicensed users (secondary users) must, according to FCC's rules, sense the spectrum before transmitting and evacuate immediately when a licensed user (primary user) appears in the same band. The most important and commonly seen primary users in white space are TV towers and wireless microphones, which have the priority over any secondary users.

While such shared-style spectrum accessing increases the efficiency of spectrum utilization, it introduces a new type of attack: primary user emulation (PUE) attack [1]. In such attack, an adversary transmits signals whose characteristics emulate those of primary users, thereby causing legitimate secondary users to erroneously identify the attacker as a primary user. The goal of such attacker is either selfishly maximizing its own spectrum usage or maliciously preventing other secondary users from communicating.

The detection of PUE attacks is a non-trivial task, because FCC requires "no modification to the incumbent (primary) system should be required to accommodate opportunistic use of the spectrum by secondary users" [2]. This implies that primary signals just stay what they were; secondary users must rely on their own to sense primary signals and differentiate emulation attackers. In other words, conventional approaches, such as embedding signatures in primary signals or employing an interactive protocol between a primary user and secondary users, cannot be applied to defend PUE attacks.

Several advanced approaches have been proposed for mitigating PUE attacks [1], [3] − [6]. However, all of them focus on the attackers that emulate stationary primary users (TV towers). They are based on the fact that the locations of TV towers are fixed and assume that these locations are pre-known by secondary users.

In the white space, wireless microphone is another important category of primary user. Since they are not stationary, the existing solutions cannot be applied. Detecting emulation attacks of mobile primary users (wireless microphones) is a much harder problem.

In this paper, we proposed a novel method to detect wireless microphone emulation attacks. In our approach, each secondary user is equipped with an acoustic sensor. Correlations between energy level of RF signal and acoustic information received by the sensor are exploited to verify the authenticity of wireless microphones.

To the best of our knowledge, this is the first work dealing with emulation attacks of a mobile primary user. In addition, our method does not require complex hardware. We demonstrate the effectiveness of our approach through extensive real-world experiments. It shows that our method achieves both false positive rate and false negative rate lower than 0.1.

The rest of this paper is organized as follows. Section Ⅱ discusses related work. Section Ⅲ states the problem, and Section Ⅳ presents our method detecting mobile PUE attacks. Section Ⅴ evaluates our work in real-world settings. Section Ⅵ concludes the paper.

II. Related Work

Chen, et al. first proposed two location based approaches to detect PUE attacks [1]. The first is *distance ratio test* based on the two-ray ground reflection model. The location of primary signal source can be determined if two or more secondary users exchange their power strength measurements, which is compared with the TV tower map to judge if it is a genuine primary user. This method is very vulnerable to signal strength fluctuation. The other method is called *distance difference test*. It utilizes synchronization pulses embedded in analog TV signals to calculate the location of primary signal, which requires strict time synchronization among secondary users.

Chen, et al. also proposed another signal strength based method which is more tolerant to signal fluctuation [3] [4]. This method relies on an underlying wireless sensor network. Each sensor measures the signal strength of the primary user. Then local averaging smooth technique is applied and signal strength geo-peaks are assumed to be the location of primary transmitters.

All these methods try to locate the signal source to detect fake primary users. The locations of genuine primary users must be known a priori. Therefore, they cannot be used when primary users are mobile or their locations are unpredictable.

Jin, et al. presented a theoretical analysis on the distribution of received power, in order to differentiate an attacker from a genuine primary user [5] [6]. A Wald's sequential probability ratio test is employed to ensure a decent false positive and false negative. However, they assume that the primary users must be far away from all secondary users, and genuine primary users and emulation attackers subject to different propagation models. These assumptions do not hold for wireless microphones.

Recognizing the difficulty of detecting wireless microphone emulation attacks, the 802.22 Task Group 1 proposed the disabling beacon protocol [7] [8], which suggests transmitting a specially designed signal before starting wireless microphones. If additional information, such as signatures, is embedded into the beacon, this method can help secondary users to differentiate genuine wireless microphones from attackers. However, there are still a great number of legacy wireless microphone users. Considering the fact that most of them have not even registered their wireless microphones, we cannot expect that they will be equipped with a separate beacon device in the near future.

III. Problem Definition

A. Preliminaries

Mobile microphones are widely used in live performances, university lectures, sporting events, etc. They typically operate in VHF or UHF bands. According to FCC's regulation, they should occupy a bandwidth no more than 200kHz and the power output is limited to 250mW or less on UHF or 50mW on VHF. In practice, this value is typically 10 to 50mW due to battery life considerations.

Most wireless microphones use frequency modulation (FM) [9]. In the following methods and algorithms, FM wireless microphones are assumed for representativeness. The transmission range of wireless microphones' RF signal is usually less than 100-150 meters.

The working process of a wireless microphone is as follows. The microphone transforms the sound wave w(t) into current signal m(t), which has the same characteristics of the original sound wave: the amplitude stands for the loudness and the frequency shows the pitch. After that, the current signal is modulated by FM, and the modulated signal S(t) is sent into the air.

In the receiver end, the demodulator listens to S(t), and demodulated it into current signal m'(t). Then, m'(t) is output to loudspeakers or power amplifiers. The regenerated sound wave is noted as w'(t).

B. Attack Model and Problem Description

The emulation attacker mimics the characteristics of a wireless microphone's signal, in order to make secondary users erroneously identify it as a primary user. We assume that the attacker has full capability to emulate wireless microphones' transmission power, modulation type, bandwidth occupation, and any other characteristics of S(t).

We also assume attackers do not emit sound wave (to emulate w(t) or w'(t)), because in that way it will be very easily detected out.

The problem to be solved is differentiating emulation attackers from genuine primary users (wireless microphones). We have discussed the incapability or limitations of existing methods in Section Ⅱ. For our method, each secondary user is equipped with a sound sensor. We identify a genuine wireless microphone by exploiting the correlation between RF signals received by the secondary user and the environmental sound captured by its sound sensor (noted as w''(t)). If the signals do not pass our correlation test, an emulation attack is assumed.

Incorporating acoustic information opens a new window for small-scale, mobile primary signal detection, but there are still substantial challenges:

1) *Correlating sound to the energy level of RF signal.* This is simple for amplitude modulation (AM), but wireless microphones use FM, where the correlation is relatively difficult to exploit. We choose to use energy detection because it is fast and simple. All cognitive radio devices are able to detect signal power without hardware modification.

A straightforward alternative is to demodulate S(t) into sound, and then compare it with w''(t). However, this means we have to add FM demodulators and rebuild the secondary users' internal circuit. More important, although they all use FM, different wireless microphones have different signal format, such as mono, stereo, bandwidth, companding techniques, etc. It is very difficult for a single device to demodulate various wireless microphones from different manufacturers.

2) *Timing constraint.* The 802.22 standard draft specifies that sensors must be able to detect wireless microphone signals over a 200kHz band within 2 seconds with both false-alarm and misdetection probabilities less than 0.1. Therefore, fast detection of emulation attacks is highly desireable.

IV. Detecting Emulation Attack of Wireless Microphone

A. Correlation between Acoustic Signal and FM Power

Let w(t) be the sound wave. The microphone transforms it to current signal m(t). Ignoring the nonlinear distortion, we have:

$$m(t) = \alpha w(t) \quad (1)$$

where α is a constant. After frequency modulation, the modulated signal is:

$$S(t) = A_c \cos\left[2\pi f_c t + 2\pi k_f \int_0^t \alpha w(\tau) d\tau\right] \quad (2)$$

where A_c is the carrier amplitude, f_c is the carrier frequency, and k_f is the sensitivity of the modulator.

Without loss of generality, let αw(t) = $A_m \cos(2\pi f_m t)$, and substitute it into Equation 2:

$$S(t) = A_c \cos\left[2\pi f_c t + \frac{k_f}{f_m} A_m \sin 2\pi f_m t\right] \qquad (3)$$

Let $m_f = k_f A_m / f_m$, we have:

$$\begin{aligned} S(t) &= A_c \cos(\omega_c t + m_f \sin \omega_m t) \\ &= A_c[\cos \omega_c t \cos(m_f \sin \omega_m t) - \sin \omega_c t \sin(m_f \sin \omega_m t)] \qquad (4) \end{aligned}$$

and noting the trigonometric relationship that:

$$\cos(m_f \sin \omega_m t) = J_0(m_f) + \sum_{n\ even}^{\infty} 2J_n(m_f) \cos(n\omega_m t)$$

$$\sin(m_f \sin \omega_m t) = \sum_{n\ odd}^{\infty} 2J_n(m_f) \sin(n\omega_m t)$$

where J_n are Bessel functions of the first kind, of order n. Substituting these two expressions into Equation 4, we have:

$$\begin{aligned} S(t) = A_c J_0(m_f) \cos \omega_c t & \\ + \sum_{n\ even}^{\infty} & A_c J_n(m_f) [\cos(\omega_c + n\omega_m)t + \cos(\omega_c - n\omega_m)t] \\ + \sum_{n\ odd}^{\infty} & A_c J_n(m_f) [\cos(\omega_c + n\omega_m)t - \cos(\omega_c - n\omega_m)t] \qquad (5) \end{aligned}$$

Equation 5 shows the frequency components of S(t). Besides f_c, it also has frequency components on $f_c \pm n f_m$ ($n \in Z^+$), which are called side bands. According to the property of Bessel functions, when x goes larger, the values of $J_o(x)$, $J_1(x)$, $J_2(x)$, etc., become closer, which means more side bands significantly contribute to the power of S(t). Therefore, the power around center frequency reduces when m_f increases, because the total power contained in an FM wave is constant [10].

As we know, k_f is a constant and $m_f = k_f A_m / f_m$. We refer to the power of S(t) within $[f_c - \Delta f, f_c + \Delta f]$ (Δf << bandwidth of S(t)) as $P_{2\Delta f}$ thereinafter. Therefore, we come to our conclusion:

Lemma: $P_{2\Delta f}$ reduces when $\frac{A_m}{f_m}$ goes up, and vice versa.

This conclusion applies to any FM wireless microphone, no matter it is mono or stereo, with or without companding.

B. Emulation Attacker Detection

As derived in Section ⅣA, $P_{2\Delta f}$ is related to both the frequency and amplitude of sound wave. Between them, the amplitude is the key factor. That is because the frequency range of human voice is from 300 to 3000Hz, where the upper bound ten times the lower bound, but loudness varies more. A normal speaker has the sound level about 40 to 80dB, and lower than 20dB when pauses. The power of 80dB is 10^6 times higher than that of 20dB, and 1000 times larger in terms of amplitude.

Therefore, loosely speaking, larger is the A_m, smaller is the $P_{2\Delta f}$.

Now we describe our method for emulation detection. If a wireless microphone signal is detected, a secondary user immediately performs the operations as follows to determine if it is an emulation attacker.

First, the secondary user shrinks its radio bandwidth to $2\Delta f$, with the center frequency unchanged (the same as the center frequency of the band where the wireless microphone is detected). $2\Delta f$ is set to 25kHz by default. Because most wireless microphones have a RF bandwidth of 100kHz or 200kHz, too large Δf makes it hard to capture the power change around center frequency. On the other hand, smaller Δf can improve the performance of our method, but some cognitive radio devices may not have such narrow band-pass filters.

Then, the secondary user synchronizes the acoustic signal with the RF signal, and samples w''(t) with its sound sensor. Every Δt, it calculates the average amplitude of the samples. At the same time, it measures the average $P_{2\Delta f}$ every Δt, and evaluates the correlation between $P_{2\Delta f}$ and averaged amplitudes. The algorithm is given as follows:

```
score = 0;
lastAvgAmp = lastAvgPwr = 0;
diff = signalSync();
wait(diff);
for (i=0; i<$n$; i++)
    time = getCurrentTime();
    wait ($\Delta t$);
    avgAmp = average $|A_m|$ from $time$ to $time + \Delta t$;
    avgPwr = average $P_{2\Delta f}$ from $time - diff$ to $time - diff + \Delta t$;
    if (lastAvgAmp != 0)
        index = fix (ln (avgAmp / lastAvgAmp) / ln $\beta$)
        if (index * (avgPwr – lastAvgPwr) < 0)
            score ++;
        else score = score – |index|;
    lastAvgAmp = avgAmp;
    lastAvgPwr = avgPwr;
if (score < 0) an attacker is assumed
```

Algorithm 1. Emulation attack detection

signalSync() is the function that synchronizes the RF and acoustic signals. Because the speed of sound is much slower than that of radio wave, they do not reach the secondary user simultaneously. In this function, we make use of pauses in human voice. When a pause occurs, the amplitude of sound wave will suddenly drop below 5 in PCM coding (8 bit sampling, max amplitude is 128), and for RF signal, all the power will concentrate at the central frequency (max $P_{2\Delta f}$ is achieved). Two signals are synchronized by sensing these sudden changes. *signalSync* returns the latency of the acoustic signal.

fix() is to round towards zero, and Δt is set to 80ms, which is restricted by our experiment equipment. n and β are adjustable parameters. n is the number of the testing rounds. The larger n, the more accurate our method is, but the longer the detection takes. β determines the algorithm's sensitivity of amplitude fluctuation.

In the algorithm, only dramatic changes (greater than β times) of A_m take effect, in order to tolerate the fluctuation of f_m. *score* is the evaluation of the correlation between A_m and $P_{2\Delta f}$. If they vary following the lemma in Section ⅣA, one point is gained, otherwise, a penalty is made. If the lemma is not followed when the A_m change is extremely large (|index| > 1), the penalty is heavier correspondingly. At the end of the algorithm, if *score* is less than zero, an emulation attack is reported. The complexity of this algorithm is linear; it spends most of its time waiting for sampling and *signalSync*.

V. EVALUATIONS

In this section, we conduct real-world experiments to evaluate our method for emulation attack detection.

A. *Experiment Settings*

We use wired microphones connected to laptops to collect environmental sound (acting as sound sensors). The raw data collected from sound card is 44.1kHz and PCM coded. Each sample is an 8-bit unsigned integer. We transform the samples into the range of [-128, 127] and average every four consecutive samples as the input of our algorithm.

We utilize an Agilent E4405B spectrum analyzer to measure the power of RF signals, whose minimum sweep time is 80ms. So in all the following experiments, Δt is set to 80ms.

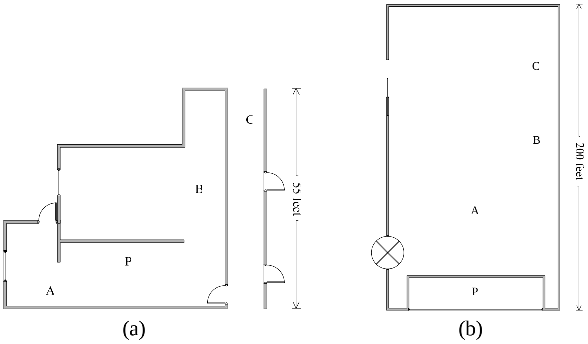

Figure 1. Layout of experiment environment

Two wireless microphones are used in our experiments. One is working on VHF band (171.9MHz) and the other on UHF (629.5MHz). Both have a bandwidth of 200kHz and 10mW power output. Their receiver ends are connected to a pair of ordinary loudspeakers (80watt).

We conduct our experiments in both crowded rooms and a spacious hall, as shown in Figures 1a and 1b, respectively. The loudspeaker connected to the primary user is put at P. Secondary users are located in A, B and C, respectively and their experiment results are averaged unless otherwise specified.

B. *False Positive and False Negative*

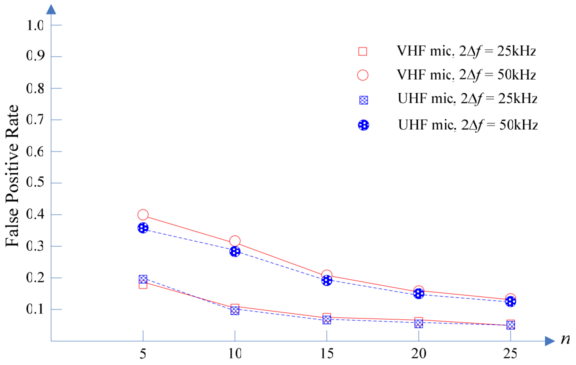

Figure 2. False positive rate

We first evaluate the false positive rate of our method. False positives refers to the detection results erroneously taken a genuine primary user as an attacker. In the first experiment, we fix the amplitude sensitivity β to e (≈2.718, we will vary β later), and test false positive rates by varying rounds n from 5 to 25 and with different $2\Delta f$ values (50kHz and 25kHz).

The result is shown in Figure 2. y-axis shows the false positive rate, which equals false positives divided by total number of tests. x-axis is the number of rounds (n). For each point in the plot, 240 tests are performed (40 at each location in Figure 1) and various voice samples are tested.

From the figure we can see that two wireless microphones act very similar. The performance of $2\Delta f = 25$kHz is much better than that of $2\Delta f = 50$kHz. In both cases, the performance of our method gets better when n becomes larger. When $2\Delta f = 25$kHz and $n \geq 15$, the false positive rates are less than 0.1, and it is as low as about 0.06 when $n = 25$.

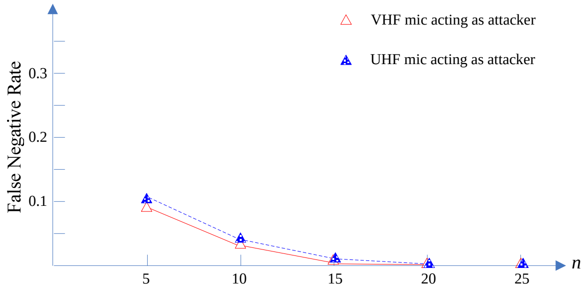

Figure 3. False negative rate

Figure 3 shows the false negative rate of our method. False negatives are the cases where an emulation attacker appears but our method fails to report. In this experiment, wireless microphones are acting as attackers, with the receiver end disconnected to loudspeakers. We eliminate w(t) by using line-in as the input of wireless microphones. That is to say, attackers emulate wireless microphones' RF signal perfectly, but without emitting sound. Here $2\Delta f$ is set to 25kHz. Other settings are the same as the false positive experiment.

From the figure we can see the false negative rate of our method is very low. As long as n is larger than 15, there is almost no false negatives.

As mentioned in Algorithms 1, β determines the sensitivity of the amplitude fluctuation of sound wave. The larger is β, the less is the sensitivity. In the next experiment, we examine the impact of β on the false positive and false negative.

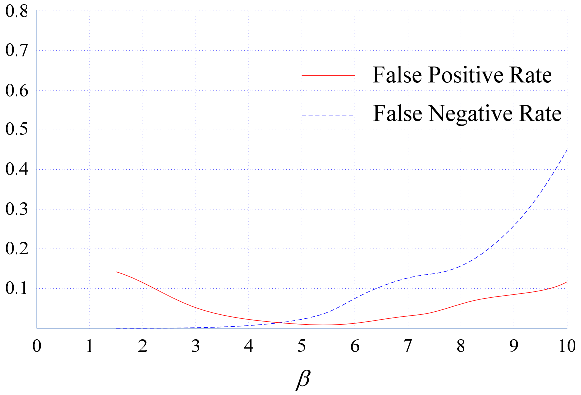

Figure 4. Impact of β on false alarm and mis-detection

In figure 4, we set n = 25 and $2\Delta f$ to 25kHz. β varies from 1.5 to 10. The solid line indicates the false positive rate while the dashed line refers to the false negative rate. The results of two wireless microphones are averaged.

From Figure 4 we observe that the false negatives increase quickly when β approaches 10. That is because when β is very large, the sensitivity becomes very low and the algorithm can hardly extract any fluctuation of sound amplitude, which causes no penalty and also no awards for *score*. In our algorithm settings, an attacker is not reported when *score* = 0. On the other hand, when β is very small, the false positive rate is relatively high. The reason lies in that high sensitivity makes the algorithm incapable of tolerating noise and the fluctuation of f_m. Considering all factors, $\beta \in (3, 6)$ is acceptable.

From the three experiments above we can conclude that when n = 25, $2\Delta f$ = 25kHz and $\beta \in (4, 5)$, our method can achieve both false positive rate and false negative rate lower than 0.05. We set $\beta = 4$ in the following experiments.

C. Dectection Time

The detection time of our method contains two parts: the execution time of *signalSync* and ($\Delta t * n$). The time of *signalSync* can be further divided into two parts: time waiting for the first RF pause and the delay of acoustic signal (*diff*). It is easy to bound *diff*. The operating range of wireless microphones is usually less than 100 meters, and the speed of sound is about 340 meters/sec. Hence, *diff* should be less than 0.3 second. For the other part, we perform experiments to measure the time waiting for the first pause. We test various sound materials, including news reports, lectures, talk shows, etc. The average value is 1.43 seconds per pause. Therefore, the total time of *signalSync* should be less than 1.7 seconds. This is consistent with our measured result in Algorithm 1, which is about 1.5 seconds. On the other hand, from previous experiments, we observe that n should be larger than 15 in order to achieve good performance. Therefore, the total detection time of our method is approximately 3 seconds (assuming Δt = 80ms).

However, Δt, which is restricted by our experiment device, can be largely reduced. The spectrum analyzer we use (Agilent E4405B) scans 400 points within the channel to calculate signal power. The way how it works makes Δt considerably large. For comparison, an 802.11 device can measure RSSI within 1ms. Therefore, the detection time of our method can be reduced to less than 2 seconds if proprietary devices are used.

D. Spatial Attenuation

In this subsection, the attenuation of RF signal and sound wave are compared. The loudspeaker connected with the primary user is turned to medium volume, which is about 102.5dB measured one meter away. RF power of two wireless microphones are tested (both power output are 10mW). Experiment results are shown in Figure 5. We can see that the power of RF signals have a significant drop at about 15 meters, which is due to multipath and body absorption.

An important observation is that when the distance gets close to 100 meters, the power of RF signals quickly drops under -70dBm, and is not detectable at 110 meters. However, the sound level of w'(t) is still about 60dB at the same distance. This experiment shows that the sound level of w'(t) decreases more slowly than the power of wireless microphone's RF signal. In other words, as long as the RF signal of wireless microphones is detectable, acoustic information, on which our method relies, is always available.

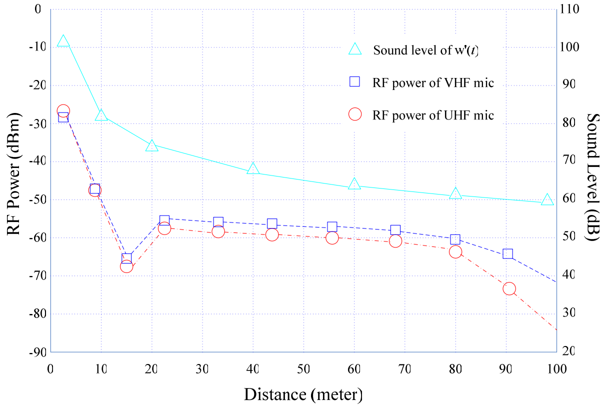

Figure 5. Attenuation comparison between RF and acoustic signal

VI. Conclusion

In this paper, we proposed a novel method to detect emulation attacks of mobile primary users. The correlation between the RF signal and acoustic signal are exploited to differentiate attackers from genuine wireless microphones.

We conducted real-world experiments to evaluate our method. The results demonstrate that our method can achieve both false positive rate and false negative rate lower than 0.1 within 3 seconds. The detection time can be further reduced when proprietary white-space devices are available.

References

[1] R. Chen and J M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *Proc. of IEEE Workshop on Networking Technol. for Software Defined Radio Networks*, pp. 110-119, Sep 2006.

[2] Federal Communications Commission, "Facilitating opportunities for flexible, efficient, and reliable spectrum use employing spectrum agile radio technologies," ET Docket No. 03-108, Dec 2003.

[3] R. Chen, J M. Park, and J H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Jl. on Sel. Areas in Commun.: Spl. Issue on Cognitive Radio Theory and Applications*, vol. 26, no. 1, pp. 25-37, Jan 2008.

[4] R. Chen, J M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *Proc. of IEEE Conf. on Comp. and Commun. (INFOCOM)*, pp. 1876-1884, Apr 2008.

[5] Z. Jin, S. Anand, and K P. Subbalakshmi, "Detecting primary user emulation attacks in dynamic spectrum access networks," *IEEE Intl. Conf. on Commun. (ICC)*, Jun 2009.

[6] Z. Jin, S. Anand, and K P. Subbalakshmi, "Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing," *ACM Mobile Comput. and Commun. Review: Spl. Issue on Cognitive Radio Technologies and Systems*, 2009.

[7] Sensing performance with the 802.22.1 wireless microphone beacon, IEEE 802.22-09/0068r1, Mar 2009.

[8] Z. Lei and F. Chin, "A reliable and power efficient beacon structure for cognitive radio systems," *IEEE Intl. Conf. on Commun. (ICC)*, May 2008.

[9] E. Reihl, "Wireless microphone characteristics," IEEE 802.22-06/0070r0, May 2006.

[10] M J. Ryan, M. Frater, "Communications and information systems," *Argos Press*, Australia, 2002.