Towards Practical Spectrum Permits

Ana Nika*, Zengbin Zhang†, Ben Y. Zhao*, Haitao Zheng*
*Department of Computer Science, University of California, Santa Barbara, CA
†Google, Mountain View, CA

***Abstract*—As spectrum is being released from legacy technologies, reforms in policies and regulations promise to spur wireless growth by distributing spectrum dynamically across wireless networks matching their traffic demands. However, a major obstacle to its adoption remains. There is no effective solution to protect licensed users from *spectrum misuse*, where users transmit without properly licensing spectrum, and in doing so, interfere and disrupt legitimate flows to whom the spectrum is assigned (and sold). In this article, we discuss an initial step towards enforcing dynamic spectrum allocation using the concept of *spectrum permit*, where authorized spectrum users embed secure spectrum permits into data transmissions, enabling patrolling trusted devices to detect devices transmitting without authorization. We highlight the development of spectrum permits, and describe *Gelato*, a spectrum misuse detection system that minimizes both hardware costs and performance overhead on legitimate data transmissions. We implement *Gelato* using USRP2 with laptops and lower cost RTL-SDR devices with smartphones. We show that both implementations are robust against attacks and can reliably detect spectrum permits in real time.**

***Index Terms*—Dynamic spectrum access, spectrum permits, spectrum misuse detection, cognitive radios**

I. Introduction

[1] Dynamic spectrum access is the clear solution in a world where innovative wireless technologies are stifled by a shortage of available spectrum caused by spectrum lock-in to legacy systems. A well-designed dynamic allocation scheme allows cognitive radio devices to obtain spectrum matching their demands while avoiding interference with peers. Towards this goal, recent research has built the core algorithms and techniques necessary for the deployment of dynamic spectrum networks in various frequency regions [13], [28], [35], [42], [46]. Particular emphasis has been given to algorithms that maximize spectrum utilization through highly efficient, short-term, local spectrum auctions [15], [39], [47]. By letting realistic short-term demands dictate the size, duration and spatial coverage of spectrum allocations, this approach enables practical and efficient use of spectrum.

A major obstacle, however, remains on our path to adopting current proposals of dynamic spectrum networks. As devices that share the same spectrum continue to grow, traditional approaches will not scale and it will be hard to detect and resolve all spectrum misuse attempts that might occur. Thus far, policy makers and researchers have not been able to find a practical and readily applicable solution to the problem of *spectrum misuse*. Specifically, we must allow users who have spectrum licenses to transmit, while preventing unauthorized users from transmitting and interfering with authorized transmissions. Unfortunately, an application, using today's cognitive radios, can easily transmit on frequencies outside of its allocated range, either accidentally due to bugs or misconfiguration, or intentionally to avoid paying spectrum license costs. Unchecked, interference from these "misuse" events will disrupt legitimate transmissions. Without effective protection, users have no assurances their transmissions would operate without interference, and would have no incentive to pay for this type of spectrum access.

Intuitively, there are two general approaches to address spectrum misuse. The first approach uses *per-device prevention* [31] to directly prevent each radio from accessing unauthorized spectrum. The enforcement module can be built into the radio hardware [41], or placed in the kernel and user space of the radio software [7]. Given the power and flexibility of cognitive radios, however, studies assert that a per-device prevention mechanism would be costly and difficult to perfect [10]. This is particularly true when the allocation of spectrum varies over time, *e.g.* when spectrum is allocated in small time segments. Furthermore, attackers can modify software and firmware to bypass any enforcement modules. Subsequent advances on both sides can lead to an arms race between designers and attackers.

A second approach is to *detect spectrum misuse* in real time, so that they can be terminated or circumvented. Yet distinguishing misuse attacks from legitimate spectrum licensees is challenging because wireless signals exhibit complex patterns and attackers can easily modify radio transmission or signature to mimic those of legitimate transmissions. In addressing these challenges, existing works have proposed solutions that rely on dense deployments of spectrum sensors, which would record local RF signal measurements, along with a device identifier for each transmission [22], [41]. The unique per-device identifiers are used to distinguish licensed users from unauthorized users. These approaches have two significant limitations. First, they require a dense deployment of costly spectrum sensors for any geographic area using this system. This is because radiometric signatures can change over time and space [5], and it is very difficult to maintain and distribute per-device identifiers without a dense sensor deployment. Second, per-device unique identifiers could become insecure, as hardware and MAC addresses can be forged, and even intrinsic hardware signatures can be replicated given the right equipment.

In this article, we introduce a new direction on real-time spectrum misuse detection, which combines the concept of

[1]Extended version of the paper "Enforcing Dynamic Spectrum Access with Spectrum Permits" published in MobiHoc'12.

spectrum permit with probabilistic attack detection. Initial efforts have led to *Gelato* [44], a robust spectrum permit system which performs real-time verification of spectrum license and reliable detection of spectrum misuse events under diverse conditions. We have implemented *Gelato* using both USRP2 with laptops and RTL-SDR devices with smartphones. Our evaluation shows that both implementations are robust against attacks and can reliably detect spectrum permits in real time. Overall, we believe a successful spectrum permit system will help pave the way for wide-spread adoption of dynamic spectrum networks.

II. The Case for Spectrum Permits

The choice of our design came from an effective analogy between the spectrum misuse problem and the problem of enforcing vehicle speed limits on roads and highways. Building a speed control into each vehicle would be difficult and costly, but a selective detection and punishment scheme in the form of highway patrols can be a very effective deterrent against speeding. Another similar problem is deterring illegal car parking, where authorization to park is dependent on the specific time and geographic location. Instead of a costly and complex per-vehicle solution, parking patrols (*e.g.* meter-maids) provide a much lower-cost and more practical deterrent.

Similarly, we believe a probabilistic system that detects and punishes unauthorized transmitters is the approach most likely to succeed in practice. The solution should avoid prohibitively high hardware costs, such as those from densely deployed spectrum sensors [22], [41], and per-device identifiers or signatures, which can be duplicated with sophisticated hardware [41]. Like highway patrols or meter maids, our solution involves a number of trusted mobile devices that patrol transmission areas to detect unauthorized users. Authorized users display time-varying one-time keys that are easily verified but cannot be duplicated. Once an unauthorized transmission is detected, trusted devices can use secure localization techniques [21], [36] to locate the misbehaving devices and stop the unauthorized transmissions.

Based on these observations, we propose a new system for securing dynamic spectrum transmissions through detection of spectrum misuse. When a user purchases a license to transmit on a given spectrum frequency, at a specific time and location, it receives from the spectrum owner a *spectrum permit*, a secure sequence of keys that prove its authorization to transmit in its operating spectrum. Users "display" their valid spectrum permits by embedding them inside the data transmissions during each time window. Trusted *police devices* patrol transmission areas, scan different spectrum ranges to passively detect each transmitter's permit, and verify its validity in real time. They can then detect misbehaving devices whose transmissions lack the necessary spectrum permits. These police devices can be used by any entities, e.g. network providers, that have legitimate concerns and seek to protect their spectrum rights.

As we will show in the following sections, a well-designed spectrum permit system will provide several advantages over prior solutions. First, permits are simple to read and verify, thereby simplifying and reducing the cost of the detection infrastructure. Second, permits are implemented as one-time, cryptographic keys. As a result, they are tamperproof, and not vulnerable to attacks leveraging sophisticated hardware.

III. The Gelato Spectrum Permit System

We now describe Gelato, our proposed *spectrum permit* system for dynamic spectrum networks. The idea is that an authorized user of a spectrum range receives a secure key that allows it to generate valid permits for a fixed time period and a specific location. In Gelato, each user broadcasts its valid spectrum permit once during each time window. Mobile "spectrum police" nodes can scan different spectrum ranges, passively listen to each transmitter's permit, and verify its validity in real time with the help of an online spectrum allocation server.

The Gelato system consists of two key components, a *permit authentication* mechanism that generates and authenticates spectrum permits at the application layer, and a *permit embedding and detection* mechanism at the physical layer that allows each user to broadcast its valid spectrum permit in its physical transmissions, and each police device to reliably detect and decode permits without decoding actual data packets. In the following, we present the permit authentication design and leave the detailed description of permit embedding and detection to Section IV.

Permit Authentication. The spectrum owner runs an online spectrum allocation database on a trusted server. It allocates spectrum in small time blocks of fixed-size T_{int}. Given a geographic location, time and frequency range, each allocated spectrum is associated with a secret K_n that represents the tail of a secure, one-way hash chain [20].

Our license verification scheme uses a secure one-way hash chain scheme, similar to authentication mechanisms used for broadcast authentication [27]. When a user U is allocated a spectrum range for n time blocks from t_0 to t_{n-1}, it is given a secret K_0. The user then computes a chain of hash codes by applying a secure one-way hash (*e.g.* SHA-1) recursively n times, producing:

$$K_0 \xrightarrow{SHA-1} K_1 \xrightarrow{SHA-1} K_2 \xrightarrow{SHA-1} \cdots K_{n-1} \xrightarrow{SHA-1} K_n$$

Starting at time t_0, the user U transmits key K_x on the embedded control channel, where x is a counter starting from n-1 that decrements once per time block. That is, the keys are transmitted sequentially in time in reverse order of the one-way hash chain, $K_n, K_{n-1}, \cdots, K_1, K_0$. Since the one-way hash function SHA-1 cannot be reversed, a node can only generate K_i from K_{i-1}. This means that attackers cannot generate valid keys for successive time windows using past key observations.

To verify the authenticity of a transmitter, a police node uses its location, time and spectrum range of the observed transmission to obtain from the database a hash chain tail K_n and a start time t_0. It computes the number of time blocks elapsed since t_0 to get the current index x of the hash chain.

Assuming the key sent on the Gelato channel is K_x, the police node applies the SHA-1 hash recursively n-x times to generate the rest of the chain. If the final result matches K_n, it proves the transmitter knows K_0, and is therefore authorized to transmit on this spectrum, location and time.

An authorized user U transmits its key K_x once per time block. Since the key can be copied and retransmitted by any nearby device, an observing police node will only consider the first transmission of K_x as valid. Even if U is not transmitting, an attacker cannot replay a previously used key K_r, because K_r does not match the correct key in the hashchain corresponding to the current time block. A police node can detect a replayed key K_r, because the number of hashes between K_r and K_n does not match the number of time blocks between the current time and t_{n-1}.

IV. Permit Embedding & Detection

We now discuss how Gelato devices physically embed (and decode) the secret keys $\{K_x\}$ of each spectrum permit into their data transmissions. This process needs to meet three key requirements: a) the permit must be flexible enough to specify a license for a given location, time and spectrum range; b) the permit must be intrinsically linked with the data transmission; and c) the permit must be readable by other devices without having to decode the data.

One straightforward solution is to transmit permits on an out-of-band control channel. It, however, suffers two disadvantages. First, it usually requires an extra radio, leading to higher hardware cost and complexity. Second, the out-of-band transmission makes it highly difficult to associate spectrum permits with data transmissions. Upon detecting a permit is being transmitted by a legitimate user, an attacker can transmit comfortably on the radio frequency covered by the permit.

In Gelato, our solution is to build on a technique in the wireless physical layer called *Cyclostationary Features*. A transmitter embeds license stream into the data transmission by controlling where it inserts the cyclostationary features. The result is visible to any police device that can sense signals on the transmitter's frequency, without decoding data content on the frequency. And more importantly, the spectrum permit is intrinsically linked to the data transmission, reflecting the actual spectrum usage.

A. Background on Cyclostationary Features

A cyclostationary signal $x(t)$ is a digital signal whose autocorrelation function is periodical in t for any time lag [33]. This property manifests into unique features in the frequency domain – a signal peak at a specific location in $x(t)$'s spectral coherence function (SCF). By capturing the RF signal of the transmitter, an external device can compute the SCF map and extract a feature at a specific *cyclic frequency* α and *spectral frequency* k, noted by (α, k). Using the prevailing OFDM communication scheme, we can intentionally introduce a cyclostationary feature into a digital signal by organizing its symbols [33]. Each OFDM symbol consists of N frequency subcarriers. We select w contiguous subcarriers indexed from p to $p+w-1$, and repeat their signals at subcarriers indexed $p+D$ to $p+D+w-1$. This new arrangement generates a group of w contiguous peaks in the SCF map at locations (α^*, k^*):

$$\alpha^* = D, \quad k^* = p + D/2 + i, \quad i = 1, 2, ..., w \tag{1}$$

Thus using a set of subcarrier repetition parameters (w, D, p), we can produce a distinct cyclostationary feature as a vertical strip of width w, centered at position ($\alpha = D$, $k = p + W/2 + D/2$). Figure 1(a) illustrates a sample feature generated at ($\alpha = 64$, $k = 102$). In this paper, we assume all Gelato transmitters use the same w/N. The peak strength s of the vertical strip depends on the received signal to noise ratio (SNR) of the data packet:

$$s = \frac{SNR}{1 + SNR} \tag{2}$$

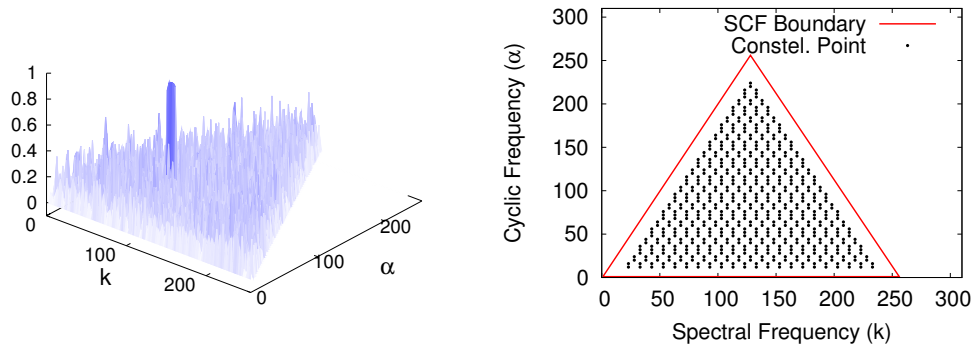
Cyclostationary features can be decoded using standard signal processing techniques without demodulating and decoding data packets. There are two algorithms that can be used for efficient cyclostationary feature detection: the FFT Accumulation Method (FAM) and the Strip Spectral Correlation Algorithm (SSCA) [29]. In the FAM method, an N-point sliding FFT followed by a downshift in frequency to baseband is used to estimate complex demodulates. Time smoothing is performed using a P-point FFT, where P depends on the cyclic frequency. In the SSCA algorithm, the complex demodulate of one of the signals is computed in the same way as in the FAM method. The final signal is smoothed in time by using an N-point FFT.

To detect cyclostationary features, each receiver computes the discrete SCF map from raw OFDM symbols and locates feature peaks [9]. It computes the correlation between the SCF map and an ideal peak pattern, producing a new SCF map. This step eliminates noise in the system, as well as random occurrences of cyclostationary property in the packet data itself. Using the new SCF map, we can easily detect the feature location (α^*, k^*) by detecting peak on the projected cyclic and spectral frequency domain. Each feature needs to be transmitted continuously for a period of time (by a group of OFDM symbols). This is to ensure that the receiver can build a stable characterization of the SCF map, and suppress the impact of frequency-selective multipath fading [32]. Therefore, in Gelato, each data packet carries a single feature to maximize its robustness.

While injecting cyclostationary features requires modifying OFDM subcarriers, the decoding process can be made completely transparent to normal data transmissions. Each receiver can first detect and extract the feature, and proceed with data decoding by ignoring all subcarriers that have been identified to carry redundant data as part of the feature.

B. Embedding Spectrum Permits

The goal of Gelato is allowing each transmitter to display a stream of its spectrum permit bits as cyclostationary features. Thus the permit is intrinsically linked to its data transmission



(a) SCF Map of a Feature (b) Sample Feature Constellation Map

Fig. 1. Building and encoding cyclostationary features. (a) A cyclostationary feature at ($\alpha = 64$, $k = 102$). (b) A sample feature constellation map for transmitters with α resolution = 4 and k resolution = 10 (FFT = 256, CP length = 64), mapping to 9 bits per feature.

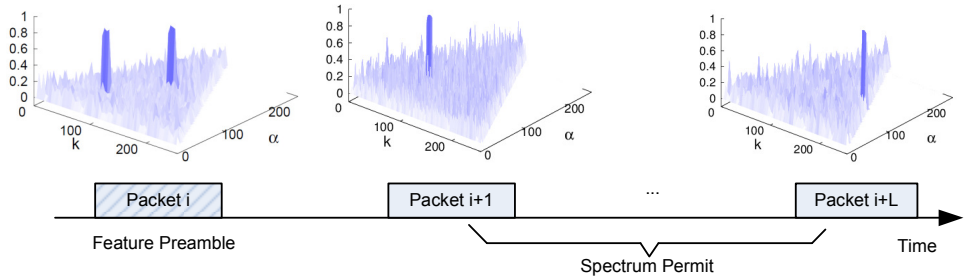


Fig. 2. Each Gelato permit consists of its feature preamble and a group of features carrying the permit bit stream. The preamble carries two features, one on each half of the k-axis, carrying information on the FFT size and CP length required to decode the subsequent feature packets.

and readable by police devices without decoding data. To do so, Gelato faces two key challenges. First, we need an effective method to convert any arbitrary permit bit streams into features. Second, the content carried by each feature depends on the underlying OFDM configuration, which can differ across devices. Police devices must first obtain the configuration in order to decode the permit.

Gelato addresses these challenges via two novel solutions. First, it builds a *feature constellation map* to associate each bit pattern with a feature peak location. Second, it introduces a *feature preamble* to "broadcast" the OFDM configuration, therefore bootstrapping the permit decoding process.

A Feature Constellation Map. We encode bit patterns by associating a given bit pattern with a feature at a specific SCF map location. Here, we refer to the collection of feature locations as the *feature constellation map*, and each location as the *feature constellation point*. Figure 1(b) illustrates a sample feature constellation map. To decode the feature, the receiving device first locates the feature peak from the SCF map, then computes encoded data as the bit pattern associated with the constellation point closest to the detected location.

The number of bits a feature can carry depends on the total number of distinct constellation points that can be reliably distinguished on an SCF map. This depends on the resolutions in the spectral frequency (k) and cyclic frequency (α) domains, *i.e.* the minimum spacing between adjacent constellation points to make them uniquely separable at the feature detector.

A Feature-bootstrapping Preamble. Embedding bit patterns into data packets is not enough to produce a signaling channel to embed spectrum licenses. We face an additional challenge: different transmitters can encode their data using very different parameters, *i.e.* FFT size and Cyclic Prefix (CP) length, both of which must be known to define a feature constellation map.

Our solution is to introduce a feature preamble carrying the transmitter's FFT size and CP length to bootstrap the receiver. Figure 2 shows an example where each spectrum license message of size M is split across a group of $L + 1$ ($L = M/n$) data packets. We embed inside the first packet of the sequence a feature preamble that "broadcasts" the FFT size and CP length. Each of the next L packets carries a n-bit cyclostationary feature.

The preamble must be decodable by all devices regardless of their OFDM configuration, and easily distinguished from normal spectrum license signal features. We encode the preamble as a set of two features, as shown in the SCF map of Figure 2. First, to make them even more easily distinguishable from normal spectrum license features, we make the width of preamble features twice the normal size. Second, we observe

that common OFDM systems use a very limited number of FFT and CP length configurations, which can easily be represented by 4-5 bits. The two features in the preamble represent values for the FFT size and CP length, and the position of each feature is associated with a particular value for that parameter. For example, the feature on the left can represent one of four possible FFT sizes (64, 128, 256, 512), by dividing the left half of the SCF map into four quadrants and assigning a value to each quadrant. Knowing the values associated with each quadrant, the receiver can determine the FFT size by looking at the relative position of the left feature on the SCF map.

C. Decoding Spectrum Permits

Ideally, the basic feature decoding method (described in Section IV-A) should be sufficient to extract features. Yet in practice, Gelato must address several challenges to implement a robust permit decoding system. These include the lack of synchronization between transmitters and police devices in both frequency and time domains, and the frequency-selective fading [12] that attenuates frequency subcarriers differently. If not addressed, both artifacts lead to significant feature decoding errors. Gelato devices remove these artifacts by reliably tracking transmission boundaries in both frequency [43] and time domains, and by tracking channel fading profiles in the frequency domain. Given the space limitations, we refer the readers to [44] for a more complete description of the practical challenges and the corresponding solutions.

V. Defending Potential Attacks

So far we have designed Gelato to provide the intended properties in the absence of adversaries. In this section, we examine potential adversarial attacks and describe Gelato mechanisms to address and detect each type of attacks.

Threat Model. We define "attackers" to include both users who transmit without license either by accident or misconfiguration, and users who do so intentionally to avoid the costs of spectrum licenses, possibly modifying their software defined radios in the process. In either case, we assume attackers' data traffic resemble legitimate transmissions, but can be altered to avoid detection.

Attackers in our model have these properties. First, each attacker has full control of its software defined radio, and can use it to eavesdrop on legitimate transmissions and transmit arbitrary data. Second, they can tune parameters such as transmission power and operating frequency, but are limited by device hardware constraints, *e.g.* finite transmission power. Third, attackers have reasonable resource limitations that prevent them from computationally revealing the secret keys, *i.e.* they cannot break strong cryptography via brute force. Finally, police nodes are mobile devices, do not transmit data, and cannot be found or compromised by attackers.

A. The Copycat Attack

To use spectrum without a permit, attackers can eavesdrop on a legitimate transmission, extract its spectrum permit, and then attempt to use the permit for its own data transmissions. This attack is relatively easy to detect, since each legitimate user only transmits her permit once during each time block. The police node can easily detect an attacker if the same permit is transmitted twice.

Within the allocated geographic area for a given permit, there might be regions where the legitimate transmission signal is weak, and the copycat transmission will go undetected. However, since each spectrum allocation request is for a given usage area, such regions are likely small compared to areas where both transmissions overlap, and the attacker can be detected as police nodes move around the area.

B. The Free-rider Attack

This attacker hides behind legitimate users, *i.e.* by sending data packets in parallel without embedding spectrum permits. If the interference from the attacker is moderate, a casual observer would only observe a legitimate permit and a single transmission formed by the union of the legitimate transmission and the free-riding transmission.

Gelato police nodes can detect this attack by comparing the signal strength of the embedded control features to the raw received signal strength to detect the contribution of hidden free-riders. If the raw signal strength is significantly higher than the signal strength observed on the control features, then one or more hidden transmitters are close by. To detect this, Gelato offers a tool that estimates the received signal strength of a transmitter from the peak strength values of its features. Specifically, Gelato estimates the signal strength S^* of a transmitter from its feature strength s,

$$S^* = \left(\frac{1}{\rho/s - 1}\right) \cdot N_0 \tag{3}$$

where N_0 is the thermal noise power, and $\rho \leq 1$ is a device-dependent parameter, *e.g.* 0.9 for the USRP2 radios and 0.99 for RTL-SDR devices that we use to prototype Gelato. If S^* is less than the raw signal strength beyond a threshold, we claim a free-rider is present.

We address frequency-selective fading by utilizing the fading profile observed by the police node. We compensate the overall signal strength estimate by a factor that depends on the channel response of subcarriers observed by the police node [44].

C. The Bad-mouth Attack

Another type of intelligent attackers can seek to "bad-mouth" a legitimate user, *i.e.* frame an innocent user to look like she is transmitting illegally. The attack can be performed by "replacing" the victim's features with false ones. Specifically, the attacker occasionally transmits one or more false features at high power in parallel to the legitimate transmissions, which overpower and override the legitimate features. The police node would only observe replacement bits, thus corrupting the legitimate permit.

Gelato police can detect the presence of bad-mouth attacks by comparing the observed raw signal strength and the one

estimated from the detected feature. In order to overpower the legitimate feature, the attacker receive power must be no less than that of the legitimate user. Thus if the observed permit is false, and the raw signal strength is occasionally more than twice the average feature-estimated strength, then a bad-mouth attacker is likely to be present.

VI. Implementation

We implemented a Gelato prototype on USRP2 GNU Radios and on smartphones connected to RTL-SDR devices. More specifically, we implement a Gelato transmitter on USRP2 radio and use RTL-SDR and USRP2 devices as receivers for normal data communications. Also, police nodes can be either USRP2 or RTL-SDR devices and can be used to verify spectrum permits and detect attackers. While we chose GNU Radios and RTL-SDR devices for their availability, our design can be ported to other platforms [11], [34], [37].

Smartphone + $20 RTL-SDR. We use as Gelato receiver or police node a platform that consists of a commodity smartphone and an inexpensive Realtek dongle (RTL-SDR for brevity) [1]. The RTL-SDR connects to the smartphone via a USB cable, behaves as a spectrum sensor and collects raw I/Q samples; while the smartphone acts as a "data processor", translating the raw data into a data stream by extracting spectrum permits. We pick RTL-SDR because of its low cost (<$20), portability (<2oz weight), wide availability, and superior frequency coverage — it operates in 52–2200MHz with a sample rate up to 2.4MHz, and transfers raw I/Q samples to the connected host on the fly. We also built an Android app to run real-time measurements and permit detection, by specifying frequency range, sampling rate and time duration.

Gelato Transmitter & Receiver. Each Gelato transmitter consists of two processing paths: the normal data path and the permit displaying path. To display a permit, we modify the OFDM subcarrier mapping module in the data path to create subcarrier repetition. We implemented pilot tones following the same pilot/data ratio of WiFi. These pilot subcarriers do not follow Gelato's repetition rule, and can degrade the feature strength.

Gelato receivers are like normal data receivers, except that we add a permit detection and removal path. This is because bits from subcarriers carrying repetitive information to display spectrum permits should be removed from the data packet. Therefore, we modify each receiver to add a feature detection module. After locating the feature, the receiver's subcarrier demapping module simply removes the w duplicated subcarriers.

Gelato Police. We implement each Gelato police node as a standalone spectrum permit detector. The police reads OFDM signals and applies our proposed mechanisms to track packet boundaries and compensate for frequency offsets. We implement the proposed cyclostationary feature detection module to identify feature peaks and extract bits. The decoded permit bit stream is then validated using the proposed permit authentication process.

Gelato police nodes are much less complex compared to typical OFDM receivers. In addition to not performing packet demodulation/decoding, they require no synchronization in time and frequency. Both are among the most complex blocks in typical OFDM receivers. We show in Section VII that Gelato police can decode features reliably without any FFT symbol level synchronization.

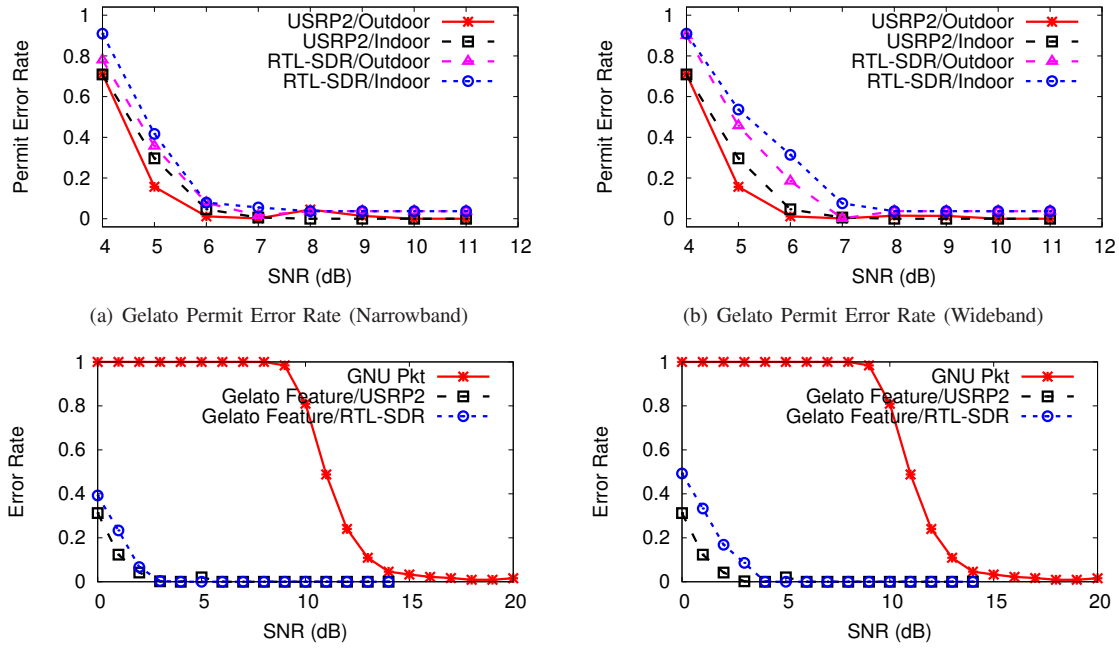
Wideband Transmissions. For robustness, wideband transmitters embed cyclostationary features over the entire band. Our current implementation uses wide bands of 6MHz, e.g. TV whitespace channels. We choose this bandwidth because our solution is suitable for TV whitespaces where FCC requires that all (high power) devices transmit identifying information conforming to a standard, allowing observers to recognize the device [8]. Later, in Section VII-C, we will examine RTL-SDR performance for other bandwidths.

While USRP2 can sense up to 25MHz, to identify wideband transmitters, RTL-SDRs need to detect these wideband features by "stitching" multiple adjacent frequency observations together. We can split each 6MHz band into 3 sections of 2MHz each; let RTL-SDR hop across the sections sequentially and aggregate the results. This is feasible since RTL-SDRs have a small frequency switching delay (<50ms) [25]. Specifically, after monitoring each 2MHz section and build the corresponding SCF map, the RTL-SDR concatenates these maps in frequency to build a wideband SCF map for wideband feature detection. This requires the transmitter to transmit the same wideband feature for at least a time period long enough to complete a single scan. For example, for a 6MHz band the transmitter should transmit the feature for $\sim$150ms.

VII. Experimental Evaluation

We evaluate Gelato using the aforementioned prototype implemented using USRP2 GNU radios and RTL-SDRs with smartphones.

Experiment Setup. For each experiment, we build a set of 1600 permits, each 160-bit long. We embed each permit into a set of 18 randomly generated data packets. Each packet contains 32 OFDM symbols and carries a single cyclostationary feature. We also inject random gaps between packets. We focus on two representative indoor/outdoor scenarios in our experiments: complex indoor environments with furniture and walls, and outdoor environments with surrounding buildings, where both experiments are performed on our university campus. To examine the impact of channel fading, we also experiment with static/mobile scenarios: a static scenario where devices were placed statically, and a low-mobility scenario where we walked around the room with the feature receiver at a normal pedestrian speed. For both scenarios, there were random human movements throughout the experiments. Finally, while our prototype supports various transmission configurations on transmit power, FFT size and CP length, we observe in our experiments that these configurations lead to similar



(a) Gelato Permit Error Rate (Narrowband)

(b) Gelato Permit Error Rate (Wideband)

(c) Gelato Permit vs. Data Transmission (Narrowband)

(d) Gelato Permit vs. Data Transmission (Wideband)

Fig. 3. Reliability of Gelato spectrum permits. (a) Gelato permits achieve a less than 5% permit error when the SNR is greater than 6dB for narrowband and less than 7dB for wideband transmissions for both USRP2 devices with laptops and RTL-SDR devices with smartphones. (c) Gelato's feature detection is much more reliable than packet decoding.

conclusions. Thus in the following we only show the results for 256 FFT and 1/4 CP length.

Our evaluation seeks to answer several key questions:

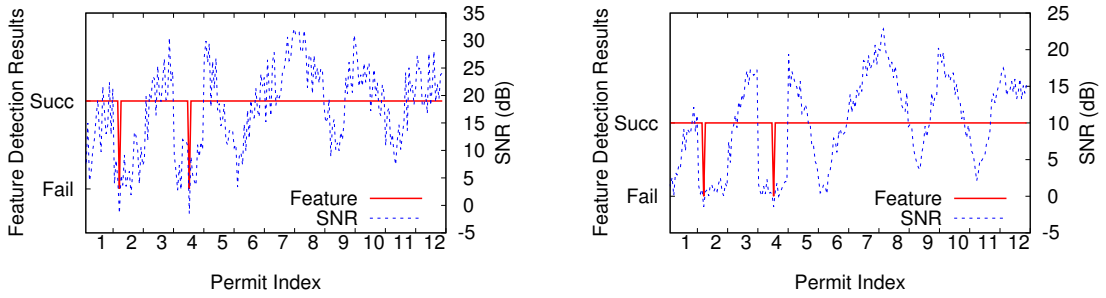
- *Can Gelato permits serve as a reliable method to authenticate spectrum usage in the presence of channel impairments and interference*? *Are the detection results similar for RTL-SDRs and USRP2 receiver/police nodes*?
- *How wideband transmissions compare to narrowband*?
- *Will Gelato's feature transmissions be more reliable than data packet reception, so that they stay transparent to data transmissions*?
- *Can Gelato police detect the presence of attackers, using the proposed feature-based signal strength estimation*?

A. Spectrum Permit Reliability

Permit Error Rate. We first examine Gelato's permit error rate under different wireless transmission profiles. Specifically, we consider narrowband (2MHz) and wideband channels (6MHz). Figures 3(a-b) shows the error rate of Gelato permit reception. Since each permit is delivered by multiple features, it can only be successfully retrieved if *all* the features are received correctly. First, in low SNR regions (SNR$\leq$ 6dB for narrowband and SNR$\leq$ 7dB for wideband), the performance in indoor environments is slightly worse than outdoors, because of the negative impact of frequency selective fading. When subcarriers carrying features suffer deep fades, the feature peak weakens and becomes hard to detect. Therefore, the error rate is higher than that of feature detection. Second, RTL-SDRs have slightly higher error rate than USRP2, especially for wideband transmissions. This happens because RTL-SDRs need to stich together several bands to detect a feature. The impact of switching delay and fading for low SNRs makes it more likely to corrupt the signals and, thus, not allow the correct feature detection. Overall, we see that the error rate reduces to <5% when the SNR exceeds beyond 7dB. For outdoor WiFi access points, this requirement typically maps to 200-300 meters of detectable range from the police node to the transmitting access point [30]. This result implies that Gelato police might need to move around a legitimate user to get a "clearer" view of its permit.

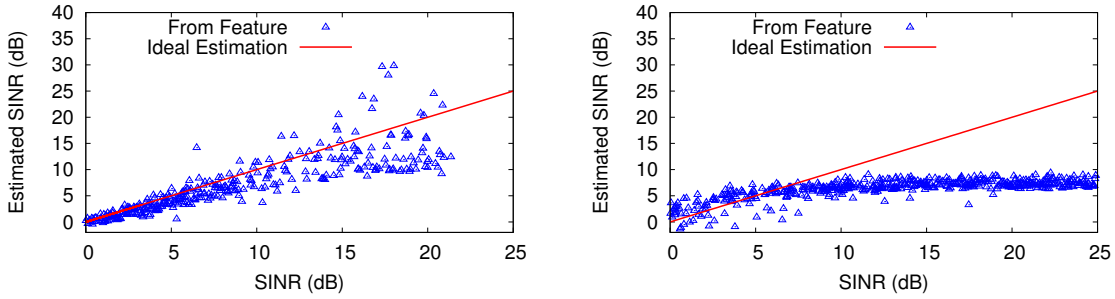
Impact on Data Transmission. A key requirement (and advantage) of Gelato is to guarantee that data transmission will not be affected by the permit display except the expected throughput loss due to subcarrier repetition. To do so, the intended receiver of each data packet needs to detect the cyclostationary feature embedded in the data packet, and uses the corresponding subcarrier repetition pattern to correct the subcarrier demapping, *i.e.* removing the repeated subcarriers. This requires that the feature decoding is at least as robust as the packet decoding at each intended data receiver.

To verify this requirement, in Figures 3(c-d) we plot the feature decoding error compared to the packet decoding error for the packets containing no features, both implemented using USRP2 radios and RTL-SDRs. For a fair comparison, we ignore feature errors caused by inaccurate packet locking, because it also prevents packet reception. Thus the corresponding feature error rate is better than that in Figures 3(a-b). Overall, we see that Gelato's feature detection is much more robust



(a) Mobile USRP2 with laptop (b) Mobile RTL-SDR with smartphone

Fig. 4. Impact of mobile police nodes. When walking around a large 12m×7m room with a Gelato receiver, we observe very few feature decoding errors caused by deep channel fades.



(a) USRP2 Signal Strength Estimation (b) RTL-SDR Signal Strength Estimation

Fig. 5. Estimating the signal strength of a legitimate user from its feature strength.

than packet decoding. Again, feature detection for wideband transmissions using RTL-SDRs has higher errors in low SNR regions, due to the impact of switching delay and channel fading.

Mobile Police Nodes. To capture the impact of police mobility, we carried the police nodes (USRP2 with laptop and RTL-SDR with smartphone) and walked around to generate a low-mobility scenario. We used the same configuration as the above static experiments and repeated it 10 times. We found that mobility has very little impact on Gelato. For example, after sending 12 permits (216 features), only two features that suffer very low SNRs were not decoded, leading to 2 corrupted permits (shown in Figure 4(a-b)). We believe that this can be compensated by adding a low-level of error-correction coding [6] redundancy into each spectrum permit. Both RLT-SDR and USRP2 devices behave similarly. Also, we observe that USRP2 SNR is about 10-12dB larger than RTL-SDR SNR as shown in prior work [25].

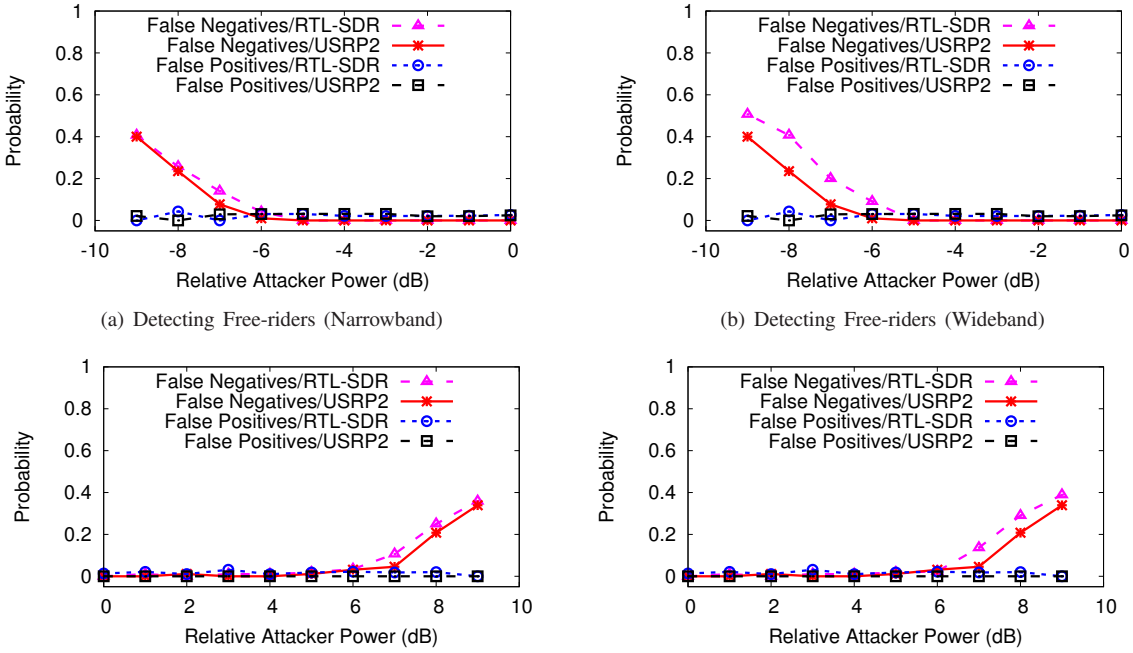
B. Attack Detection

Next, we examine Gelato's ability to detect adversarial attacks. Since reliable permit transmissions and verification already enable the detection of copycat attacks, we focus on examining free-riders and badmouth attacks.

Accuracy of Feature-based Signal Strength Estimation. Since Gelato detects attacks by comparing the observed signal strength with the feature-estimated signal strength, we first verify the proposed signal strength estimation. To explore the impact of channel noise and interference (from other transmitters or attackers), we activate another transmitter to inject interference to the police node in the presence of the legal transmitter's transmission, and record the SINR observed at the police node.

Figures 5(a-b) compare the estimated signal strength with the true value. We see that the estimation is quite accurate when the SINR is less than 8dB, but the accuracy drops at larger SINR values for both USRP2 and RTL-SDR devices. This is due to the non-linear mapping between the SINR and peak strength. At high SINRs, a small deviation in peak strength computation manifests into larger errors in the estimated signal strength. Furthermore, we observe that the estimated SINR for RTL-SDRs is capped by 5dB for larger SINR values. This happens, because RTL-SDRs have limited sensing sensitivity and feature peaks for high SINR values will be estimated as they have smaller strength.

Attack Configuration. We implement both attacks and vary the attacker power to emulate different physical distance or power profile. Our experiments consists of an attacker, a legitimate transmitter (victim) and a police node. For both attacks, we use the *Relative Attacker Power* $(S_A(dB) - S_V(dB)$ to capture the difference between the received power of the attacker $S_A(dB)$ and that of the victim observed at the police node $S_V(dB)$. Because the legitimate receiver can be at any location within the legitimate transmitter's coverage area, as the police node moves around the network, the relative attacker power it observes also reflects the one observed at



(a) Detecting Free-riders (Narrowband)

(b) Detecting Free-riders (Wideband)

(c) Detecting Badmouth Attackers (Narrowband)

(d) Detecting Badmouth Attackers (Wideband)

Fig. 6. Performance of Gelato's attacker detection.

the legitimate receivers. The higher the relative attacker power observed at the victim receiver, the higher the performance degradation to the legitimate transmissions.

Detecting Free-riders. Figures 6(a-b) shows that in indoor settings Gelato can reliably detect almost all (95+%) of free-riders whose signal strength is no more than 6dB weaker than the legitimate user. This means that the attacker needs to transmit at a very low power level to evade the detection, thus producing much less harmful interference to the legitimate user. Detecting weaker attackers is less reliable due to increased errors in feature based signal strength estimation at high SINRs. The presence of a weak attacker only leads to a small drop in feature peak strength, which could also be caused by random noise and interference. This ambiguity increases false negatives (or miss detections). RTL-SDRs have similar detection rate as USRP2s for narrowband transmissions, but false negatives increase for wideband transmissions. Our outdoor experiments observe a slightly degraded accuracy (80% detection rate), because outdoor transmissions suffer higher temporal variations from dynamic surroundings such as vehicles passing by, which introduces additional noise to feature peaks.

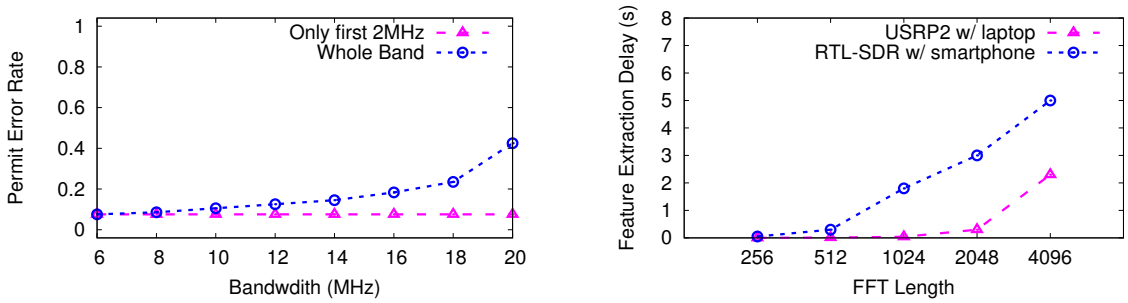
For both scenarios the rate of false positives remains insensitive to attacker power settings. This is because false positives are mainly caused by the use of pilot tones which degrades feature peak strength and leads to false alarms. The impact depends on pilot locations rather than attacker power, and thus remains constant throughout the experiments. We repeated the experiments using frequency-selective fading scenarios and the results (omitted for brevity) are similar.

Detecting Bad-Mouthers. To overwrite the victim's feature, a bad-mouther must transmit false features at a sufficiently high power. Figures 6(c-d) show the performance of detecting bad-mouth attacker as a function of the attacker's relative power level for indoor scenarios. We see that Gelato's attack detection is highly effective – it forces the attacker to transmit at a significantly higher power (6+dB over the victim) in order to evade detection. These high-power attacks, however, are more visible and can be easily detected by checking signal strength and data transmission consistency over space and time, such as those proposed by [40]. Finally, we observe similar trends on false negatives and false positives like those of the free-rider attacks. Similar results occur for scenarios with frequency-selective fading.

C. Overcoming RTL-SDR/Smartphone Limitations

So far, we have performed our evaluation using 6MHz wideband signals. We observed in Figure 3(b) that the RTL-SDR permit error rate increases when SNR$\leq$ 7dB, due to switching delay. We now analyze how RTL-SDR performs for larger bandwidths. We fix the SNR to 7dB and we derive the permit error rate for several wideband signals (from 8-20MHz). Figure 7(a) shows that the error rate increases very little for signals up to 14MHz, while it becomes significant for wider bands. One potential solution is to use only the subcarriers that belong to the first 2MHz to transmit each feature. Thus, RTL-SDR devices will not need to hop across bands and stich the bands together. In such a case, the permit error rate remains stable when the bandwidth increases (as shown in Figure 7(a)).

Secondly, in order to have a practical permit system, RTL-



(a) RTL-SDR Permit Error Rate for Wide Bands

(b) Feature Extraction Computation Delay on Smartphones

Fig. 7. RTL-SDR performance: (a) The RTL-SDR permit error rate increases when the bandwidth increases. (b) The computation delay on smartphones is small for small FFT sizes.

SDR devices need to detect features very fast. However, feature detection requires FFT computations, which in general is slow, especially if performed using smartphones. We have optimized our Android application to perform FFT and other computations as fast as possible. Figure 7(b) shows the delay in feature detection using USRP2 with laptop and RTL-SDR with smartphone. We observe that for small FFT lengths the delay is small and it increases when the FFT length increases. Even with increased FFT length, the application will need only few seconds to detect a permit, thus, RTL-SDRs can be used effectively as police nodes.

Finally, since each RTL-SDR device draws power from the smartphone via the USB connection, we need to understand if this will be a problem for our system. A recent study [4] has shown that the total power draw depends on the specific tuner chip used in each RTL-SDR. There are two popular RTL-SDR models, the Rafael Micro R820T dongle that draws up to 1.2Watt while the FC0013 dongle draws about 0.6Watt [4]. In our study, we used the Rafael Micro R820T dongle and we evaluated the phone's power consumption when the RTL-SDR device is attached using the Monsoon Power Monitor [2]. We identified that the smartphone consumes about 1.5Watt when the dongle is attached and no other activities run on the phone. This number is slightly higher than that reported by [4], likely due to the difference in RTL-SDR manufacturers (although the devices use the same tuner type). However, the power consumption of the dongle is on par with the power draw of the LTE (1.5Watt) radio on smartphones when operating in the receiving mode [14], [26] and it can be further reduced if we use the more energy-efficient FC0013 model.

VIII. Related Work

Spectrum Authentication and Misuse Detection. Existing work has examined *per-device prevention* where spectrum misuse is enforced per-device in order to prevent devices from operating without a valid spectrum license [3], [7], [24], [31], [41]. Other approaches are based on *external monitoring and detection* and the solutions are designed for different network contexts. In the context of opportunistic spectrum access that contains primary and secondary users, prior works can authenticate each primary user using its unique link transmission characteristics created via a "helper" node [23], detect extra (illegal) transmitters by examining received signal strength [22], or apply extensive signal measurements to locate each transmitter and comparing their locations with those of legitimate users to identify violators [5]. These solutions require dense and costly deployments of monitoring sensors and helpers, and often assume ideal propagation models. More importantly, they place the burden of misuse detection completely on the detection infrastructure, making it costly and highly complex to perfect. Gelato takes a different direction - by forcing legitimate users to display their spectrum permits, Gelato shifts the responsibility to the users, significantly reducing the complexity and cost of the detection infrastructure.

Similar to Gelato, recent approaches propose using the physical layer for spectrum authentication. SpecGuard [17] is motivated by Gelato and outsources spectrum misuse detection to mobile users. The authors claim they do not use cyclostationary feature detection because it will not be applicable in a system based on mobile devices. However, we show in our work that mobile devices are able to detect and decode cyclostationary features with high accuracy. [19] uses a controlled amount of inter symbol interference in the transmitted pulses that can utilized to embed the authentication signal. [18] embeds the authentication information into the transmitted waveform by inserting an intentional frequency offset. In SafeDSA [16] a user embeds her spectrum authorization information into the cyclic prefix of each physical-layer symbol. However, none of the studies validates the proposed solution in practical settings under different scenarios and attacks. Our wok shows in detail that the Gelato system is practical and performs well using different devices in real-life scenarios.

Signal Embedding. Research efforts in this area have developed strategies to embed "side" information either directly into raw data bits (*i.e.* digital watermarking), or into physical-layer signals [38], [45]. These solutions all require demodulation/decoding of the original data transmission, which is infeasible in our scenario.

Gelato is motivated by prior work on cyclostationary features [33], but applies the concept in the context of displaying spectrum permits within transmissions. Unlike prior work,

Gelato proposes a novel feature constellation map that allows features to carry arbitrary control information, and a robust detection framework to decode features in the presence of transmission artifacts and attacks.

IX. Conclusion

In this paper we tested Gelato, a new robust spectrum permit system for authenticating spectrum usage and detecting misuse. Gelato devices transmit spectrum permits as cyclostationary features embedded inside their data transmissions, while trusted police devices patrol transmission areas to detect misbehaving devices. Gelato permits are reliable and "universally" decodable without requiring packet decoding. Detailed testbed experiments show that Gelato is a feasible, practical and cost-effective method for enforcing spectrum rights. As new spectrum access policies grow in adoption around the world, it is clear that a system like Gelato will be essential to ensure correct spectrum usage.

References

[1] http://sdr.osmocom.org/trac/wiki/rtl-sdr.
[2] https://www.msoon.com/LabEquipment/PowerMonitor/.
[3] Brik, V., Shrivastava, V., Mishra, A., Banerjee, S., and Bahl, P. Towards an architecture for efficient spectrum slicing. In *Proc. of HotMobile* (2007).
[4] Brouwers, N., and Langendoen, K. Will dynamic spectrum access drain my battery? *Embedded Software Report Series, ES-2014-01* (2014).
[5] Chen, R., Park, J.-M., and Reed, J. Defense against primary user emulation attacks in cognitive radio networks. *IEEE JSAC* (Jan. 2008), 25 –37.
[6] Davey, M., and Mackay, D. Reliable communication over channels with insertions, deletions, and substitutions. *IEEE Trans. on Information Theory 47*, 2 (Feb. 2001), 687 –698.
[7] Denker, G., Elenius, D., Senanayake, R., Stehr, M.-O., and Wilkins, D. A policy engine for spectrum sharing. In *Proc. of IEEE DySPAN* (2007).
[8] FCC. Second report and order and memorandum opinion and order. *FCC-08-260* (2008).
[9] Gardner, W. A. Signal interception: a unifying theoretical framework for feature detection. *IEEE Trans. on Commun. 36*, 8 (1988), 897–906.
[10] Giacomoni, J., and Sicker, D. Difficulties in providing certification and assurance for software defined radios. In *Proc. of IEEE DySPAN* (2005).
[11] Gummadi, R., Ng, M., Fleming, K., and Balakrishnan, H. Airblue: A system for cross-layer wireless protocol development and experimentation. In *MIT Tech. Report* (2008).
[12] Halperin, D., et al. Predictable 802.11 packet delivery from wireless channel measurements. In *Proc. of SIGCOMM* (2010).
[13] Hong, S. S., and Katti, S. R. Dof: a local wireless information plane. In *SIGCOMM* (2011).
[14] Huang, J., Qian, F., Gerber, A., Mao, Z. M., Sen, S., and Spatscheck, O. A close examination of performance and power characteristics of 4G LTE networks. In *MobiSys* (2012).
[15] Ji, Z., and Liu, K. R. Belief-assisted pricing for dynamic spectrum allocation in wireless networks with selfish users. In *SECON* (2006).
[16] Jin, X., Sun, J., Zhang, R., and Zhang, Y. Safedsa: Safeguard dynamic spectrum access against fake secondary users. In *Computer and Communications Security* (2015).
[17] Jin, X., Sun, J., Zhang, R., Zhang, Y., and Zhang, C. Specguard: Spectrum misuse detection in dynamic spectrum access systems. In *INFOCOM* (2015).
[18] Kumar, V., Park, J.-M., and Bian, K. Blind transmitter authentication for spectrum security and enforcement. In *ACM SIGSAC Conference on Computer and Communications Security* (2014).
[19] Kumar, V., Park, J.-M., Clancy, T. C., and Bian, K. Phy-layer authentication by introducing controlled inter symbol interference. In *Communications and Network Security (CNS)* (2013).
[20] Lamport, L. Password authentication with insecure communication. *Comm. of the ACM 24*, 11 (Nov 1981).
[21] Lazos, L., and Poovendran, R. Serloc: secure range-independent localization for wireless sensor networks. In *Proc. of WiSe* (2004).
[22] Liu, S., Chen, Y., Trappe, W., and Greenstein, L. ALDO: An anomaly detection framework for dynamic spectrum access networks. In *Proc. of INFOCOM* (2009).
[23] Liu, Y., Ning, P., and Dai, H. Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures. In *Proc. of IEEE S&P* (2010).
[24] Mueck, M., Ivanov, V., Choi, S., Kim, J., Ahn, C., Yang, H., Baldini, G., and Piipponen, A. Future of wireless communication: Radioapps and related security and radio computer framework. *IEEE Wireless Communications 19*, 4 (2012).
[25] Nika, A., Zhang, Z., Zhou, X., Zhao, B. Y., and Zheng, H. Towards commoditized real-time spectrum monitoring. In *HotWireless* (2014).
[26] Nika, A., Zhu, Y., Ding, N., Jindal, A., Hu, Y. C., Zhou, X., Zhao, B. Y., and Zheng, H. Energy and performance of smartphone radio bundling in outdoor environments. In *WWW* (2015).
[27] Perrig, A., et al. Efficient and secure source authentication for multicast. In *Proc. of NDSS* (2001).
[28] Rahul, H., Kushman, N., Katabi, D., Sodini, C., and Edalat, F. Learning to share: narrowband-friendly wideband networks. In *SIGCOMM* (2008).
[29] Roberts, R. S., Brown, W. A., and Loomis, H. H. Computationally efficient algorithms for cyclic spectral analysis. *IEEE Signal Processing Magazine 8*, 2 (1991).
[30] Robinson, J., et al. Assessment of urban-scale wireless networks with a small number of measurements. In *Proc. of MobiCom* (2008).
[31] Sherman, M., Comba, A., He, D., and McDonald, H. A cognitive policy management framework for dod. In *MILITARY COMMUNICATIONS CONFERENCE* (2010).
[32] Sutton, P., Lotze, J., Nolan, K., and Doyle, L. Cyclostationary signature detection in multipath rayleigh fading environments. In *Proc. of Crowncom* (2007).
[33] Sutton, P., Nolan, K., and Doyle, L. Cyclostationary signatures in practical cognitive radio applications. *IEEE JSAC* (Jan. 2008), 13 –24.
[34] Tan, K., et al. SORA: High performance software radio using general purpose multicore processors. In *Proc. of NSDI* (2009).
[35] Tian, Z., and Giannakis, G. B. A wavelet approach to wideband spectrum sensing for cognitive radios. In *1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications* (2006).
[36] Čapkun, S., et al. Secure location verification with hidden and mobile base stations. *IEEE Trans. on Mobile Computing* (April 2008), 470–483.
[37] Wireless open-access research platform, http://warp.rice.edu.
[38] Wu, K., et al. Free Side Channel: Bits over Interference. In *Proc. of MobiCom* (2010).
[39] Wu, Y., Wang, B., Liu, K. R., and Clancy, T. C. A scalable collusion-resistant multi-winner cognitive spectrum auction game. *IEEE Transactions on Communications 57*, 12 (2009).
[40] Xu, W., et al. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proc. of MobiHoc* (2005).
[41] Xu, W., Kamat, P., and Trappe, W. TRIESTE: A trusted radio infrastructure for enforcing spectrum etiquettes. In *Proc. of SDR workshop* (2006).
[42] Yang, L., Hou, W., Cao, L., Zhao, B. Y., and Zheng, H. Supporting demanding wireless applications with frequency-agile radios. In *NSDI* (2010).
[43] Yang, L., Hou, W., Cao, L., Zhao, B. Y., and Zheng, H. Supporting demanding wireless applications with frequency-agile radios. In *Proc. of NSDI* (2010).
[44] Yang, L., Zhang, Z., Zhao, B., Kruegel, C., and Zheng, H. Enforcing dynamic spectrum access with spectrum permits. In *Proc. of MobiHoc* (2012).
[45] Yu, P., Baras, J., and Sadler, B. Physical-layer authentication. *IEEE Trans. on Info. Forensics and Security 3*, 1 (2008), 38–51.
[46] Yucek, T., and Arslan, H. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE communications surveys & tutorials 11*, 1 (2009).
[47] Zhou, X., Gandhi, S., Suri, S., and Zheng, H. ebay in the sky: strategy-proof wireless spectrum auctions. In *MobiCom* (2008).