SpecGuard: Spectrum Misuse Detection in Dynamic Spectrum Access Systems

Xiaocong Jin[1], Jingchao Sun[1], Rui Zhang[2], Yanchao Zhang[1] and Chi Zhang[3]
[1]Arizona State University
[2]University of Hawaii
[3]University of Science of Technology of China
{xcjin, jcsun, yczhang}@asu.edu, ruizhang@hawaii.asu, zhangchi@ufl.edu

***Abstract*—Dynamic spectrum access is the key to solving worldwide spectrum shortage. The open wireless medium subjects DSA systems to unauthorized spectrum use by illegitimate users. This paper presents SpecGuard, the first crowdsourced spectrum misuse detection framework for DSA systems. In SpecGuard, a transmitter is required to embed a spectrum permit into its physical-layer signals, which can be decoded and verified by ubiquitous mobile users. We propose three novel schemes for embedding and detecting a spectrum permit at the physical layer. Detailed theoretical analyses, MATLAB simulations, and USRP experiments confirm that our schemes can achieve correct, low-intrusive, and fast spectrum misuse detection.**

I. Introduction

Dynamic spectrum access (DSA) is the key to solving worldwide spectrum shortage. In a DSA system, the spectrum owner leases its licensed under-utilized spectrum to unlicensed users. To improve the spectrum efficiency, the spectrum owner can regulate the spectrum access by issuing spectrum permits with each specifying a frequency channel, a geographic area, and a time duration [1]. A valid spectrum permit serves as an authorization to use the corresponding frequency channel in the specified area and time duration.

The open wireless medium subjects DSA systems to *spectrum misuse*. Specifically, illegitimate users without proper spectrum permits can still use the spectrum freely. In the presence of spectrum misuse, legitimate users having paid for valid spectrum permits will experience severe interference and thus may be discouraged from further using DSA systems; the spectrum owners without sufficient legitimate users will have no incentives to deploy and operate DSA systems. This situation calls for effective mechanisms to detect spectrum misuse to unleash the full potential of DSA technology.

How can we detect spectrum misuse in DSA systems? Consider a typical DSA communication session with a transmitter and a receiver. An intuitive solution involves the transmitter sending its spectrum permit along with its data traffic. The spectrum permit can be verified by a third node which is referred to as a *misuse detector* hereafter. If the spectrum permit is designed to be unforgeable based on cryptographic techniques, an authentic spectrum permit proves legitimate spectrum use. If an invalid or no spectrum permit is detected, the misuse detector reports to the spectrum owner who can take further actions to physically locate the illegitimate transmitter and then possibly apply law enforcement.

A sound realization of the intuitive solution above is very challenging and must satisfy three basic requirements.

- *Correct*: False-positive and false-negative rates should be low enough. A false positive (negative) here refers to a legitimate (an illegitimate) user mistaken for an illegitimate (a legitimate) user.
- *Low-intrusive*: The impact on legitimate communications should be very small. This implies little or no modification to the receiver's protocol stack, negligible negative impact on its reception capabilities, and also very little effort at the transmitter.
- *Fast*: Spectrum misuse should be quickly detected. There are two implications. First, there should be a misuse detector around the DSA transmitter with overwhelming probability. A promising approach is to explore mobile crowdsourcing by recruiting ubiquitous mobile users as misuse detectors. Second, the time to verify the spectrum permit should be very short.

There have been a few attempts to detect spectrum misuse in DSA systems. The first approach assumes a tamper-proof transceiver to prevent unauthorized spectrum access [2]–[4], but such trusted transceivers are very difficult or expensive to build and can also be hacked by capable attackers. The second method relies on a dedicated sensor network which is very costly and difficult to deploy and maintain [5]. A more recent method, Gelato [1], requires every legitimate spectrum user to embed a cryptographic spectrum permit into its physical-layer cyclostationary-features, which can be opportunistically verified by dedicated misuse detectors dispatched by the spectrum owner. Since there cannot be too many dedicated misuse detectors due to cost considerations, many illegitimate users may be undetected or detected after a long time. In addition, cyclostationary-feature detection has high computational complexity and extremely long sensing time [6], which are less suitable for crowdsourced spectrum misuse detection via resource-constrained mobile users.

This paper presents *SpecGuard*, the first crowdsourced spectrum misuse detection framework for DSA systems. Motivated by Gelato, SpecGuard requires a spectrum permit to be embedded into and detected from physical-layer signals. To address the aforementioned issues that Gelato currently has, however, SpecGuard outsources spectrum misuse detection to ubiquitous mobile users and also explores more efficient customized modulation schemes than resource-demanding cyclostationary-feature detection. SpecGuard offers three schemes for different scenarios. The first scheme works when the transmitter has a relatively large freedom of transmission power control; the transmitter embeds permit bits into physical symbols by modifying original constellation points to higher power levels. This scheme incurs higher power consumption on the transmitter but no negative impact on the receiver's data reception. In contrast, the second scheme works when the transmitter is more constrained in power control; the transmitter sends permit bits by introducing smaller variations to original constellation points and also modifying them to both higher and lower power levels. This scheme incurs lower power consumption on the transmitter but possible negative impact on the receiver's data reception. Finally, the third scheme assumes that the transmitter trusts and shares the spectrum permit with the receiver; the transmitter sends permit bits through a higher-order constellation than the original at the same transmission-power level. This incurs the lowest power consumption on the transmitter and also no negative impact on the receiver's data reception. All the three schemes enable mobile misuse detectors to reliably decode spectrum permits from physical-layer signals by efficient energy detection and thus detect spectrum misuse with low false positives and negatives.

Our contributions can be summarized as follows. First, we propose SpecGurad, the first crowdsourced spectrum misuse detection framework for DSA systems. SpecGuard features three novel schemes aiming at different scenarios. Second, we theoretically show that SpecGuard can achieve correct, low-intrusive, and fast spectrum misuse detection. Finally, we confirm the efficacy and efficiency of SpecGuard by detailed MATLAB simulations and USRP experiments.

II. Additional Related Work

Besides [1]–[4], the following work is also related.

There is significant effort on mitigating false sensing reports about the presence/absence of primary spectrum users (e.g., [7]–[9]). This line of work is orthogonal to SpecGuard.

Another line of work [10]–[12] aims at testing whether the legitimate primary user is using a licensed channel. SpecGuard has a different purpose by attempting to verify whether a spectrum user has a valid spectrum permit. In [12], the primary user sends an authentication tag by shifting the phases of QPSK constellation points, and a verifier detects the tag by examining the phases of QPSK symbols and then verifies it. This scheme has also been extended to QAM in [13]. In contrast, the spectrum permit in SpecGuard is embedded differently, and we prove that SpecGuard leads to better noise resilience and shorter permit transmission time. In addition, this scheme [12], [13] is evaluated only through MATLAB simulations, and its performance in real scenarios is not revealed. By comparison, SpecGuard is evaluated through both MATLAB simulations and USRP experiments.

Additionally, Dutta *et al.* proposed to implement a covert channel [14] by embedding secret information in the physical-layer signals of wireless communication protocols. Their main goal is to ensure that the covert channel is visible to the intended receiver only. SpecGuard differs significantly from [14] in its aim and scope. In particular, the spectrum permit in SpecGuard is designed to be easily detectable by misuse detectors, and we do no attempt to hide it from anyone.

Finally, Kumar *et al.* proposed a PHY-layer authentication [15] by introducing controlled inter symbol interference to identify rogue transmitters in DSA. The P-DSA mechanism does not provide the transparency property. Thus, HM-DSA was proposed. However, practical use of the schemes is probably hindered by the high error rate of the authentication bits. Moreover, no practical USRP experiments were performed.

III. System and Adversary Models

A. System Model

SpecGuard is in charge by an operator. The SpecGuard operator can itself be a spectrum owner or profit by managing spectrum permits for multiple spectrum owners.

SpecGuard relies on mobile crowdsourcing. A recent Cisco report [16] projects that the number of mobile-connected devices will hit 10 billion in 2016, which implies sufficient geographic coverage especially in populated metropolitan areas where DSA systems are expected to play significant roles. Since DSA is expected to be pervasive in future wireless communication systems, it has been widely expected that future mobile devices can perform spectrum sensing [17], [18]. So we are motivated to use ubiquitous mobile users capable of spectrum sensing as misuse detectors in SpecGuard. The SpecGuard operator may also deploy relatively few dedicated misuse detectors as in Gelato as a complement.

Mobile users need strong incentives for joining SpecGuard. Such rewarding mechanisms as perks or badges have been proved very successful in soliciting mobile users for crowdsourcing applications. Due to space limitations, we assume the existence of such incentive mechanisms.

B. Adversary Model

We adopt the following adversary model. The illegitimate spectrum user is assumed to fully control his radio transceiver, which renders the hardware defenses in [2]–[4] inapplicable. In addition, he does not have a valid spectrum permit, so he has to use the spectrum without a permit, with a fake one, or by replaying an intercepted valid permit. Moreover, he is computationally bounded and cannot break the cryptographic primitives underlying SpecGuard. We also assume that illegitimate spectrum use lasts sufficiently long to make spectrum misuse detection meaningful. Finally, misuse mobile detectors may be compromised to report wrong detection results.

IV. SpecGuard Overview

In this section, we outline the SpecGuard operations. There are three entities involved: the transmitter (the spectrum user sending data), the misuse detector, and the receiver (the spectrum user receiving data).

A. Spectrum-Permit Construction

A spectrum permit refers to a cryptographic authorization by the SpecGuard operator to use a specific channel in a certain area and duration. To construct a spectrum permit, we make three assumptions. First, the licensed spectrum is divided into non-overlapping channels, each identified by a unique channel index. Second, the geographic region for the DSA system is divided into non-overlapping cells of equal size, each identified by a unique cell index. Finally, time is divided into slots of equal length, and all the devices are loosely synchronized to a global time server.

We adopt the efficient hash chain to construct spectrum permits. Let $h(x)$ denote a cryptographic hash function such as SHA-1 [19] applied to any input x. We also let $h^{\eta}(x)$ denote η successive applications of h to x. Every legitimate user purchases spectrum usage from the SpecGuard operator by specifying the channel index, cell index, and time duration of interest. Assume that the requested time duration consists of $\gamma \geq 1$ slots. Upon receiving the spectrum-access request, the SpecGuard operator selects a random number n_{γ} of sufficient length (say, 160 bits), recursively computes $n_i = h(n_{i+1}), \forall i \in [0, \gamma - 1]$, and finally sends n_{γ} to the legitimate user who then recursively computes $\{n_0, \ldots, n_{\gamma-1}\}$. In SpecGuard, n_i serves as the spectrum permit of the legitimate user in slot i of the requested duration. The communications between the legitimate user and the operator should be secured using traditional mechanisms such as TLS [20].

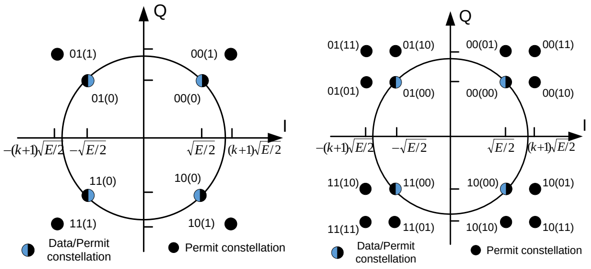
B. Spectrum-Permit Transmission and Detection

The legitimate transmitter needs to keep transmitting the spectrum permit n_i in slot i ($\forall i \in [1, \gamma]$) of the requested duration. The spectrum permit n_i is embedded into physical-layer signals by proper power control in the modulation phase, and it can be extracted by misuse detectors in the demodulation phase. The details are deferred to Section V.

C. Spectrum-Permit Verification

The SpecGuard operator activates spectrum-permit verification (or equivalently misuse detection) either according to some random schedule or when the legitimate user complains about severe interference. To do so, the SpecGuard operator chooses some misuse detectors in the specific area to ensure sufficient area coverage. It also sends the channel index, the starting time of the time duration, and the hash value n_0 to each chosen misuse detector with traditional TLS-like security mechanisms. For every slot $i \in [1, \gamma]$ of the specified time duration, each chosen misuse detector first tries to detect the ith candidate permit from the physical-layer signals on the specified channel, denoted by n'_i, and then compares n_0 with $h^i(n'_i)$. If the permit n'_i is authentic (i.e., $n'_i = n_i$), the equation $n_0 = h^i(n'_i)$ should hold; otherwise, the transmitter is very likely to be a spectrum misuser.

Misuse-detection results are reported to the SpecGuard operator. If any spectrum misuse is reported, the SpecGuard operator can dispatch some personnel to do some field test to physically locate the illegitimate transmitter and then stop spectrum misuse by possibly involving law enforcement. Finally, the SpecGuard operator rewards each misuse detector whose detection result is consistent with the field test.



(a) $M = 2$ (b) $M = 4$

Fig. 1: Constellation for Scheme 1.

V. Spectrum-Permit Transmission and Detection

In this section, we detail how spectrum permits are transmitted and detected.

A. QPSK Background

We assume QPSK as the physical-layer modulation scheme to ease the presentation, though our schemes can easily support general QAM. QPSK is a primitive modulation scheme in many applications and standards such as IEEE 802.11b, IEEE 802.11g and Bluetooth 2. It changes the phases of in-phase (*I*) and quadrature (*Q*) components separated by 90°. It uses four phases: $\pi/4, 3\pi/4, 5\pi/4$, and $7\pi/4$, corresponding to four constellation points (often called symbols) equi-spaced around a circle. We assume that the original QPSK constellation points have an amplitude of $\sqrt{E/2}$ for each component. So the energy per QPSK symbol is E.

B. Scheme 1

In Scheme 1, the transmitter continuously sends the spectrum permit for the current time slot along with its data packets. To tolerate transmission errors, we apply FEC encoding to the spectrum permit. Although there are many FEC schemes available, we choose the repetition code for its simplicity. How the repetition code is implemented depends on the constellation design discussed shortly.

1) Permit transmission: Scheme 1 embeds the permit into physical-layer symbols by modifying the original QPSK constellation. Assume that the transmitter wants to send one permit bit per data symbol. In this case, each permit bit is repeated continuously m times, where m is a system parameter. For example, if "0110" is an excerpt of the spectrum permit, it is encoded as "000111111000" for $m = 3$. If the permit bit is 0, the transmitter sends the original QPSK symbol; otherwise, it sends a new QPSK symbol by scaling the original QPSK symbol with a factor of $k+1$. Here k is a system parameter, and its impact will be analyzed in Section VI. For clarity, we show the constellation graph for Scheme 1 in Fig. 1a, where there are two permit-constellation points in each quadrant with the inner one overlapping with the original QPSK data-constellation point. The bit value in parentheses indicates the permit bit, and the two constellation points in each quadrant correspond to the same data bits but different permit bit. For example, if the original QPSK symbol is $(\sqrt{E/2}, \sqrt{E/2})$ for data bits 00, the transmitter sends $(\sqrt{E/2}, \sqrt{E/2})$ for a permit bit 0 and $((k+1)\sqrt{E/2}, (k+1)\sqrt{E/2})$ for a permit bit 1.

We can easily extend Scheme 1 to transmit two or more permit bits per data symbol by using an M-QAM constellation for permit bits, where M is a power of 2. In fact, the aforementioned scheme in Fig. 1a can be considered as a 2-QAM constellation for permit bits. An example for $M = 4$ is given in Fig. 1b, in which two permit bits are embedded in each data symbol. In this case, the permit bits are grouped into segments of $\log_2(M)$ bits, and each segment is repeated continuously m times. For example, if "011011" is an excerpt of the spectrum permit, it is encoded as "010101101010111111" for $M = 4$ and $m = 3$. Additionally, we note that it is necessary to have the data bits differentially coded to address the phase ambiguity that commonly exists in PSK or QAM modulations [21]. However, if we also apply differential coding to permit bits, it will be more difficult to decode permit bits because differential coding often produces more demodulation errors [21]. We tackle this challenge by a special coding strategy for permit bits, as shown in Fig. 1b. First, the permit symbols inside each quadrant are Gray-coded such that any two adjacent permit symbols differ only by one bit. Second, the permit symbol layout in each quadrant can be rotated 90° clockwise or counterclockwise to match the permit symbol layouts in its neighboring quadrant. In this way, in case of phase shift, although the constellation might have been rotated, the permit bits are still likely to be correctly decoded since after the phase correction, the symbols can be mapped to a constellation point with the correct coding bits except that it is in fact not the original constellation point.

A permit may be transmitted via one or multiple data packets, which depends on both the length of data packets and the constellation for permit bits. In addition, permit embedding should start right after the preamble and header of each packet are transmitted until either permit bits are all sent or all the data symbols have been used up.

2) Permit detection: In a duration specified by the SpecGuard operator, each chosen misuse detector keeps detecting a spectrum permit from physical-layer signals on the corresponding channel. Permit detection is divided into sessions, each starting right after detecting the preamble and the header of a data packet until enough permit bits are decoded to construct a candidate permit. The preamble enables synchronization and the header enables the detector to know the size of the packets whereby it knows when to prepare synchronization with the next packet. If the misuse detector misses the preamble of the current data packet, it will not start extracting the permit bits until it detects the preamble of the next data packet.

There are two possible strategies for decoding a permit bit. Assume that each data symbol carries one permit bit, corresponding to the eight-point constellation in Fig. 1a. In the hard-decision strategy, the detector finds the constellation point in Fig. 1a closest to each received symbol and then decodes the embedded permit bit as either 1 or 0. Since each permit is consecutively repeated m times, the majority rule is then applied to determine each permit bit. In the soft-decision strategy, the detector finds the constellation point which has the shortest average distance to every m consecutive symbols associated with the same permit bit. The corresponding permit bit can thus be decoded. Soft decision intuitively outperforms hard decision, which is further validated in Section VII.

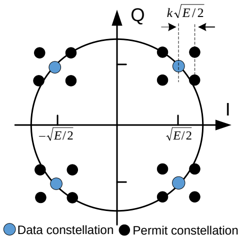


Fig. 2: Constellation for Scheme 2.

Permit transmission and detection in Scheme 1 are totally transparent to the receiver. Specifically, the receiver still performs QPSK demodulation according to the original 4-point data constellation. In addition, the increased amplitudes of the data symbols carrying permit bit 1 imply a higher SNR (signal-to-noise ratio), leading to more error-resilient data transmissions for the receiver. This aspect will be further analyzed in Section VI.

3) Transmission parameters: Scheme 1 involves four key transmission parameters: E, k, m, and M. The transmitter can easily determine E by estimating the SNR [22], [23]. According to our analytical results in Section VI, it can decide the rest parameters to make sure that the permit can be successfully detected by misuse detectors with a sufficiently high probability. Each misuse detector needs to know E, k, and m to correctly decode permit bits. This can be accomplished with the help of the SpecGuard operator. Specifically, the transmitter sends the transmission parameters via the SpecGuard operator to each misuse detector.

C. Scheme 2

Scheme 2 is motivated by the possible power constraint imposed on the transmitter in Scheme 1. In particular, the detection errors for permit bits in Scheme 1 are highly dependent on the minimum distance, i.e., $k\sqrt{E}$ for $M = 2$ and $k\sqrt{E/2}$ for $M = 4$, between permit-constellation points in the same quadrant. Given E, the larger k, the higher the transmission power, the lower the detection errors for permit bits, and vice versa. In practice, however, k cannot be too large due to many constraints. For example, FCC often imposes an upper limit on the transmission power, and the transmitter may have low energy residue. In addition, if the original constellation is higher-order QAM, the distance between adjacent constellation points may have been very small; if we use a large k to ensure low detection errors for permit bits, the errors for data bits at the receiver will increase.

We propose Scheme 2 to achieve comparable detection performance for permit bits with statistically lower energy

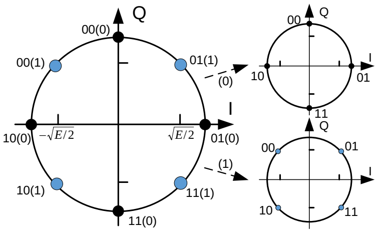


Fig. 3: Constellation for Scheme 3.

consumption at the transmitter. The key idea is to use smaller deviations from original constellation points to encode the same permits. This is achieved by increasing or decreasing the coordinates of the original constellation points according to permit bits. An example is shown in Fig. 2 with four permit-constellation points added in each quadrant, where each data symbol carries two permit bits. Note that the minimum distance between the permit-constellation points is now $2k\sqrt{E/2}$, implying lower detection errors for permit bits in comparison with Scheme 1 ($M = 4$). Assuming that the permit consists of uniformly distributed ones and zeros, the average energy level per data symbol is $(1+k^2)E$ in Scheme 2 in contrast to $(1+k+k^2/2)E$ in Scheme 1. The same rationale can be applied when the underlying modulation scheme is the more general QAM at different orders. Unlike in Scheme 1, the data reception of the receiver in Scheme 2 may be negatively affected, which will be fully analyzed in Section VI. Other operations of Scheme 2 are similar to those of Scheme 1.

D. Scheme 3

We propose Scheme 3 to further reduce the power consumption of the transmitter and also eliminate the negative impact on the receiver's data reception. Our motivation is that the data transmitter and receiver often trust each other and have bidirectional communications, so spectrum permits can be shared between them for using the same spectrum in the current communication session. Scheme 3 fully explores the receiver's knowledge about the spectrum permit and transmits the spectrum permit through a novel constellation design.

1) Permit transmission: We illustrate permit transmission in Scheme 3 still with QPSK as an example. The transmitter starts permit transmission after the preamble and header of its data packet are transmitted. The preamble and packet header are modulated with the original QPSK, but the rest data bits, when paired with the permit bits, follow the constellation in Fig. 3. After all the permit bits are transmitted, the original QPSK is reapplied to the remaining data bits. Specifically, we add four constellation points (represented by black colors) to the QPSK constellation and form a special 8-PSK constellation with the following properties.

- Each constellation point represents three bits, among which the lease significant bit (LSB) indicates a permit bit, and the others represent two data bits.
- Two adjacent constellation points have different LSBs.
- The first two bits of the four black (or grey) constellation points follow Gray coding. In other words, any two adjacent black (or grey) constellation points only differ by one bit in their first two bits.
- Each grey constellation point forms a pair with the first clockwise black point, and they differ only in the LSB. Each grey-black point pair is identified by the first two bits of the symbol value.

Scheme 3 encodes one permit per data symbol. The transmitter first determines the grey-black point pair based on the two data bits to send, and then it picks either the grey or black point based on the permit bit to transmit. For example, it sends the constellation point corresponding to the sequence 001 to convey two data bits 00 and a permit bit 1. Unlike in Scheme 1 and Scheme 2, we do not apply repetition codes to permit bits because the detection errors can be small enough due to the relatively large distance between each pair of grey and black constellation points. To further improve the error tolerance, we can append to the spectrum permit a Reed Solomon (RS) or other FEC code which is more efficient. The analysis of the error tolerance is deferred to Section VI. In addition, if a packet is not long enough to convey all the permit bits, the transmitter continues transmitting the rest of permit bits through subsequent data packets.

As in Schemes 1 and 2, phase ambiguity needs to be resolved in Scheme 3. A phase recovery error in this case will either lead to no change on permit bit decoding or only revert bit 0 to bit 1 or vice versa. Assume that the channel is slow-fading such that the same phase shift applies to the entire spectrum permit. We just let the misuse detector verify the bit-wise reverted bit sequence if the original bit sequence does not pass the verification. For example, assume that the detector obtains a candidate permit as "100110" after decoding the data symbols. If the phase recovery fails, the candidate permit will fail the verification; the correct permit should be "011001" and can pass the verification instead.

2) Permit detection and verification: Each misuse detector decodes each permit bit according to the 8-PSK constellation using the proposed coding pattern. In particular, permit decoding starts right after the detector sees the preamble and header of the data packet. Each received symbol is compared with the eight constellation points, and the LSB of the closest one tells the embedded permit bit. The detector buffers all the consecutively decoded bits and then verifies the correctness. The misuse detector reports spectrum misuse if it cannot detect a valid spectrum permit after a sufficient number of attempted verifications, which is determined by the permit error rate. Permit detection and verification cease until the detection duration specified by the SpecGuard operator elapses.

It is slightly tricky for the data receiver to decode the data bits. The receiver knows the current permit and thus can predict the next permit bit to receive. As shown in Fig 3, the 8-PSK constellation can be divided into two QPSK constellations according to the LSB (or permit bit). If the next permit bit is expected to be 0, the transmitter decodes the received symbol with the upper QPSK constellation; otherwise, the lower QPSK constellation is used. Since the distance between adjacent points in the upper and lower constellations is the same as that in the original constellation, we can expect the detection errors for data bits to be the same as in the original QPSK constellation when permit bits are not embedded. So the negative impact on the receiver's data reception can be eliminated. In addition, the energy consumption of the transmitter is the same as when permit bits are not embedded.

VI. Theoretical Analysis

In this section, we analyze the correct, low-intrusive, and fast properties of SpecGuard.

A. Correctness Analysis

The correctness of SpecGuard is analyzed. We first derive the bit error rate (BER) for the permit bits whereby to derive the false-positive and false-negative rates of the three schemes. We make the following assumptions to make the analysis tractable. The channel is assumed to be AWGN with zero mean and power spectral density $N_0/2$. Recall that E denotes the energy of an original constellation point. We define SNR as $\gamma = E/N_0$. We also assume that a spectrum permit is of L bits and is repeated m times in Schemes 1 and 2, where m is an odd integer. Finally, we assume that the detector reports a spectrum misuse when it fails to detect a valid spectrum permit in α consecutive attempts.

Since the AWGN channel does not introduce phase shift, we simply adopt non-differential QPSK modulation in the analysis. Analyses based on differential QPSK can be complicated and a closed-form solution is difficult to obtain. Hence, we assume coherent detection and perfect recovery of the carrier frequency and phase. However, as we will see in Section VII-B, in practice, these assumptions may not be valid due to various channel conditions and effects.

Theorem 1. For Scheme 1, the permit BER for $M = 2$ is

$$P_{b,1}^{M=2} \approx \mathbf{erfc}(k\sqrt{\gamma}/2)/2, \tag{1}$$

and the permit BER for $M = 4$ is

$$P_{b,1}^{M=4} \approx \mathbf{erfc}(k\sqrt{\gamma/2}/2)/2. \tag{2}$$

Proof: According to [21], the symbol error rate (SER) is approximately $P_s \approx \frac{W_{d_{\min}}}{2}\mathrm{erfc}(\frac{d_{\min}}{2\sqrt{N_0}})$, where $d_{\min}$ refers to the minimum distance between any two constellation points, and $W_{d_{\min}}$ corresponds to the number of neighbors at this distance. When $M = 2$, $d_{\min}$ equals $k\sqrt{E}$ and $W_{d_{\min}}$ equals one. So we obtain Eq. (1). When $M = 4$, $d_{\min}$ equals $k\sqrt{E/2}$, and $W_{d_{\min}}$ equals 2. Assuming that Gray coding is adopted, we can estimate the BER as half of the SER in Eq. (2). ■

Theorem 2. The permit BER for Scheme 2 is

$$P_{b,2} \approx \mathbf{erfc}(k\sqrt{\gamma/2})/2. \tag{3}$$

Theorem 3. The permit BER for Scheme 3 is

$$P_{b,3} \approx \mathbf{erfc}(\sqrt{\gamma}\sin(\pi/8)). \tag{4}$$

We then deduce the permit error rate (PER) which can be approximated by the probability when all the L permit bits are correctly extracted. As said in Section V-B2, we can use either the hard-decision or soft-decision strategy to decode a permit bit that is repeatedly transmitted m times. Due to space limitations, we only show the analysis for the hard-decision strategy and will compare these two strategies with MATLAB simulations in Section VII. Since the soft-decision always outperforms the hard-decision when the same bits are repeated, the PER for the latter can be used as an upper bound.

Theorem 4. The PER for Schemes 1 and 2 under the hard-decision strategy can be derived as

$$\begin{aligned} P_p =& 1 - (\binom{m}{\lceil m/2\rceil}(1-P_b)^{\lceil m/2\rceil}P_b^{m-\lceil m/2\rceil} \\ &+ \binom{m}{\lceil m/2\rceil+1}(1-P_b)^{\lceil m/2\rceil+1}P_b^{m-\lceil m/2\rceil-1} \\ &+ ... + (1-P_b)^m)^L, \end{aligned} \tag{5}$$

where P_b is given in Eq. (1), Eq. (2), or Eq. (3).

Since each spectrum permit is not repeated in Scheme 3, the PER of Scheme 3 is simply $P_p = 1 - (1 - P_{b,3})^L$.

Given the PER derived above, the false-positive rate can be simply estimated as P_p^{α}, and it will be evaluated with MATLAB simulations in Section VII.

A false negative in SpecGuard may happen in the following four cases when the transmitter is illegitimate.

- **Case 1:** The transmitter sends a randomly guessed permit which happens to be correct. The probability for this case can be estimated as $(1 - P_p)/2^L$. When L is sufficiently large (say, 160 bits), this probability is negligible.
- **Case 2:** The transmitter sends a randomly guessed permit which is incorrect but changed to the correct one due to transmission errors. As long as the SNR is good enough or the PER is sufficiently low, we can expect the probability for this case to be negligible as well.
- **Case 3:** The transmitter first decodes the correct permit sent by the legitimate transmitter as a misuse detector, and then it attempts to use the decoded permit for its own transmissions. In SpecGuard, each spectrum permit is valid for only one short time slot, so the illegitimate transmitter can at best use the permit in the current slot which can be set very short. In addition, the legitimate transmitter who experiences severe interference can report to the SpecGuard operator. Therefore, this case has negligible impact on SpecGuard.
- **Case 4:** All the misuse detectors are compromised by the transmitter and thus do not report spectrum misuse. Since the detectors are randomly chosen mobile users, it is very unlikely to have all of them compromised.

Hence, the false-negative rate of SpecGuard is negligible.

B. Detection Time (Analysis of the Fast Property)

Now we analyze the time it takes to correctly detect a spectrum permit. We assume that the payload of each data packet is l bytes long and transmitted at a rate of R bit/s. For simplicity, we neglect the non-payload portion of a data packet (such as the preamble and header) which is often much shorter than the payload. Then the packet transmission rate is $\frac{R}{8l}$ packets/s. Let x denote the number of data packets required to transmit a complete L-bit spectrum permit. We can easily compute x for different schemes: (1) $x = \lceil \frac{Lm}{4l} \rceil$ for Scheme 1 ($M = 2$); (2) $x = \lceil \frac{Lm}{8l} \rceil$ for Scheme 1 ($M = 4$) and Scheme 2; (3) $x = \lceil \frac{L}{4l} \rceil$ for Scheme 3. Given the PER P_p computed above, the average detection time for all the schemes is computed as $T = \frac{8lx}{R(1-P_p)}$ seconds. Examples are given in Section VII to show that SpecGuard can achieve a small T.

C. *Low-intrusiveness Analysis*

Now we analyze the data BER at the data receiver.

Theorem 5. The data BER of Scheme 1 is upper-bounded by

$$\overline{\text{BER}_{1,\text{data}}} \approx \textbf{erfc}(\sqrt{\gamma/2})/2, \tag{6}$$

and lower-bounded by

$$\underline{\text{BER}_{1,\text{data}}} \approx \textbf{erfc}(\sqrt{(1+k^2)\gamma/2})/2. \tag{7}$$

Theorem 6. The data BER of Scheme 2 is upper-bounded by

$$\overline{\text{BER}_{2,\text{data}}} \approx \textbf{erfc}((1-k)\sqrt{\gamma/2})/2, \tag{8}$$

and lower-bounded by

$$\underline{\text{BER}_{2,\text{data}}} \approx \textbf{erfc}((1+k)\sqrt{\gamma/2})/2. \tag{9}$$

Given the decoding process in Section V-D2, the data BER of the receiver in Scheme 3 is the same as in the original QPSK constellation, i.e., $\textbf{erfc}(\sqrt{\gamma/2})/2$.

VII. Performance Evaluation

In this section, we evaluate SpecGuard using MATLAB simulations and USRP experiments. We also compare SpecGuard with [12] despite their different application scenarios.

In our evaluations, we use SHA-1 as the hash function for spectrum permits, which are 160-bit long. The data packets have a constant payload length of 1,500 bytes, so a spectrum permit can be embedded into a single data packet in all three schemes. Moreover, each data point in MATLAB results is an average over 2,000 data packets, and each data point in USRP results represents an average across 10,000. It is worth pointing out that the numerical results based on our theoretical analysis in Section VI match well with our MATLAB results. We have to omit them here due to space constraints.

The key parameters in our evaluations include the channel SNR (i.e., γ), the number of repetitions for a permit bit (i.e., m), and the scaling factor of the symbol coordinates (i.e., k). According to many references such as [24], the channel SNR in [10,15), [15, 25), and [25, 40) indicates very poor, poor, and very good wireless channels, respectively. Finally, two cases in Scheme 1 ($M = 2$ or 4) are differentiated by Scheme 1.1 and Scheme 1.2 whenever necessary.

A. *MATLAB Simulations*

Fig. 4 compares the permit error rate (PER) of the soft-decision and hard-decision strategies for Scheme 1.1. We see that the soft decision outperforms the hard decision in all cases. So we focus on reporting the evaluation results based on the soft decision only due to space limitations.

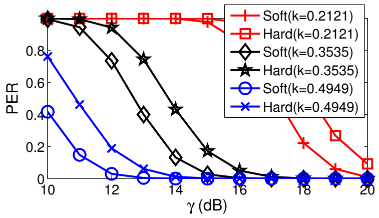


Fig. 4: Soft decision vs. hard decision.

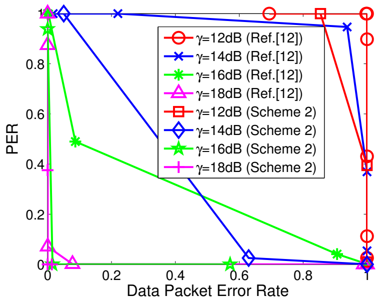


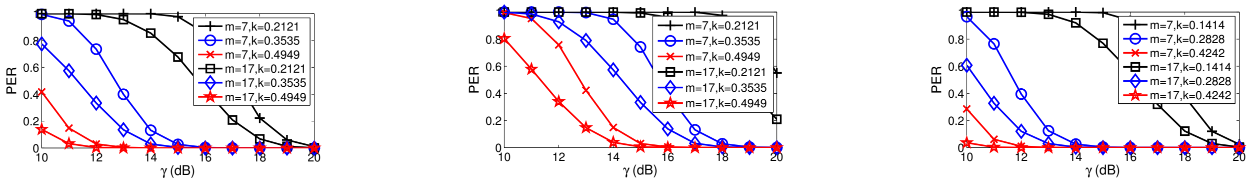
Fig. 9: Comparison between Scheme 2 and [12].

Fig. 5 shows the impact of k on Schemes 1 and 2. k ranges from 0.2121 to 0.4949 in Scheme 1 and from 0.1414 to 0.4242 in Scheme 2 to emulate tighter power constraints. As we see, the PERs of both schemes can be dramatically reduced as k increases, especially when γ is large. In addition, Fig. 5a and Fig. 5b show that Scheme 1.2 incurs a slightly higher PER than Scheme 1.1, which is consistent with the analysis in Eq. 1 and Eq. 2. We can also observe a PER reduction in Schemes 1 and 2 as m increases from 7 to 17. This is an expected benefit for using repetition codes. In general, the larger m, the lower PER, and vice versa.

We also evaluated the PER for Scheme 3 in MATLAB. When γ equals 11|12|13|14|15|16|17|18 dB, the PER is 1.00|0.99|0.92|0.66|0.31|0.07|0.02|0.00. This result highlights the superior permit detection performance of Scheme 3 in contrast to Schemes 1 and 2. One may note that all our schemes have very high PERs when $\gamma \in [10, 15]$ dB. As said above, $\gamma \in [10, 15]$ corresponds to very poor wireless channels over which normal data transmissions are unlikely to occur [24]. In other words, all our schemes have sufficiently low PERs and work well in normal channel situations.

Based on the above PER results, we further analyze the false-positive and false-negative rates of our three schemes. The false-positive rate is simply P_p^{α} (cf. Section VI-A), where α is the number of verification attempts. Fig. 6 shows the impact of α on different PERs. We can clearly see that as long as P_p is relatively small or the channel is sufficiently good, the false-positive rate of our three schemes is almost negligible. For example, when $\gamma = 16$ dB (poor channel), we have $P_p = 0.07$ in Scheme 3, leading to a false-positive rate of 0.07 for $\alpha = 1$ and 1.6×10^{-6} for $\alpha = 5$.

Moreover, we associated the results in Fig. 5 with the analysis in Section VI-B to evaluate the fast property of SpecGuard. Here we let the data-transmission rate $R = 2$ Mbit/s and



(a) Scheme 1.1

(b) Scheme 1.2

(c) Scheme 2

Fig. 5: Permit Error Rates for Scheme 1 and Scheme 2.

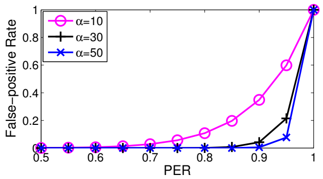


Fig. 6: False-positive rate.

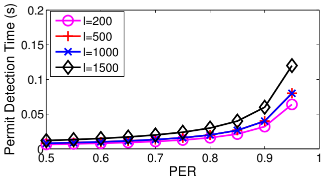


Fig. 7: Average permit detection time.

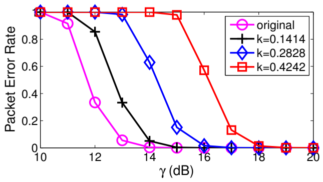
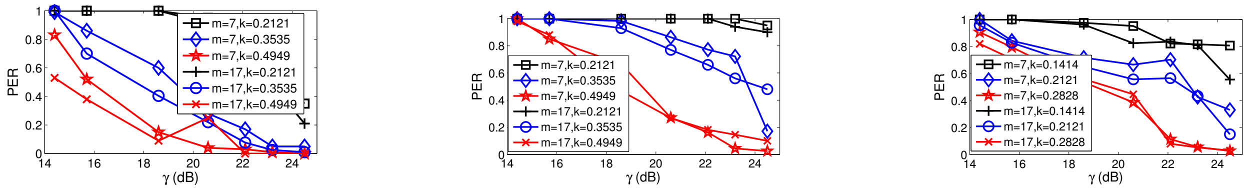


Fig. 8: Data error rate for Scheme 2.



(a) Scheme 1.1

(b) Scheme 1.2

(c) Scheme 2

Fig. 10: PER performance using USRP.

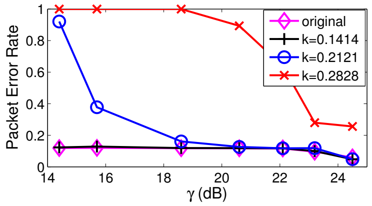


Fig. 11: Data-packet error rate for Scheme 2 using USRP.

TABLE I: The energy overhead of Scheme 1 and Scheme 2.

k	0.14	0.21	0.28	0.35	0.42	0.49
Scheme 1	15%	23%	32%	41%	51%	61%
Scheme 2	2%	4%	8%	12%	18%	24%

the repetition parameter $m = 17$. Fig. 7 shows the impact of l (data-payload length) on the average permit detection time for Scheme 1.1 and Scheme 3. Generally, the average permit detection time increases with l. In particular, larger data packets means that the time gap between the transmission of two consecutive permits becomes longer, leading to longer permit detection time. Additionally, even when the PER is very high (e.g., 0.95) and $l = 1,500$ bytes, the detection time is around 0.12 s in Scheme 1.1 and Scheme 3, indicating very fast spectrum misuse detection. We have similar results for Scheme 1.2 and Scheme 2, which are omitted for lack of space.

Furthermore, we evaluated the impact of our schemes on the data-packet error rate of the receiver. As expected, the data-packet error rate is slightly decreased in Scheme 1 because the scaling factor k effectively increases the transmission power and thus SNR. In addition, the data-packet error rate in Scheme 3 quite matches that of the original QPSK modulation, which confirms that Scheme 3 has no negative impact on the receiver's data reception. In contrast, the data-packet error rate in Scheme 2 is slightly increased, as shown in Fig. 8. Generally, the larger k, the more data-packet errors due to the reduced minimum distance between data-constellation points (cf. Fig. 2). Scheme 2 still works well for high SNRs.

Table. I reports the energy overhead for Scheme 1 and Scheme 2 as a percentage, where a spectrum permit is assumed to comprise uniformly distributed zeros and ones. Obviously, Scheme 2 always incurs low energy overhead than Scheme 1.1 and Scheme 1.2 at the cost of possible negative impact on data decoding. In contrast, Scheme 3 has zero energy overhead due to its special constellation design. It is worth pointing out that the energy overhead of Scheme 1 and Scheme 2 can still be very low to reach sufficiently low false-positive rate in normal channel conditions. For example, if Scheme 1.1 is used, when SNR is 15 dB, the PER can be around 0.7 if m is 17 and k is 0.2121. This corresponds to 23% additional energy overhead. However, since the detection is efficient, the transmitter does not need to embed the permit bits all the time, thus making the overall energy overhead a lot lower.

Finally, we jointly compared the permit and data decoding performance of Scheme 2 with the work in [12] in Fig. 9. In the comparison, we fixed $m = 7$ and varied the value of k. For [12], the shifted angle was changed from 0.1 to 0.7 rad. Generally, the closer the curves to the origin, the lower decoding errors for the permit and also the data packet, and vice versa. It is clear that Scheme 2 excels in almost all the cases. As discussed above, Scheme 2 performs generally worse than Schemes 1 and 3 when considering both PER and data-packet error rate. Therefore, all our schemes have better permit and data decoding performance than the work in [12].

B. USRP Experiments

We prototyped SpecGuard on USRP N210 with GNU Radio and placed three USRPs in a normal lab environment with furniture, computers, humans, walls, etc.. There were also human activities such as walking during the experiments. Three USRPs were separated equally with a rough distance of three meters, with each serving as a different entity in SpecGuard: the transmitter, the receiver, or the detector.

Fig. 10 shows the PER for Scheme 1 and 2, where we restricted the SNR γ between 14 and 25 dB in the experiments. Generally, the larger m, the lower PER, and vice versa. It is also clear that Scheme 1.1 is more robust in low SNR cases. Different from the simulation results, we found that the working SNR range is limited in our experiments. For example, it is somehow difficult for Scheme 2 to correctly decode the permit at an SNR lower than 14 dB. We conjecture that this difference is due to the imperfect phase recovery and AGC, multipath, frequency-selective fading, and other random channel effects. All of these factors lead to slightly worse practical performance. In real applications, the performance can be improved by better coding schemes as well as advanced techniques to mitigate those aforementioned channel effects.

Consistent with MATLAB simulations, Scheme 3 still achieves the lowest PER. When γ is 14.4|15.7|18.6 dB, the PER is 0.59|0.12|0.00; when γ is higher than 18.6 dB, the PER remains zero. These results demonstrate the high efficacy of Scheme 3 for spectrum misuse detection in practice.

We also evaluated the impact of our three schemes on the data-packet error rate. In contrast to the original QPSK modulation, our results confirmed that Scheme 1.1 and Scheme 1.2 both can slightly lower the data-packet error rate, and Scheme 3 has almost no impact on the data-packet error rate. We are more concerned about Scheme 2's negative impact on the data transmission. As shown in Fig. 11, a large k may not be feasible in low SNR cases for Scheme 2, due to frequent data-packet errors. Scheme 2, however, can still work very well in high SNR cases with a small k.

VIII. Conclusion

In this paper, we proposed SpecGuard, the first crowdsourced solution to detecting spectrum misuse in DSA systems. SpecGuard provides three different schemes for mobile detectors to detect and verify a spectrum permit from physical-layer signals of a target transmitter. Detailed theoretical analysis, MATLAB simulations, and USRP experiments have confirmed that SpecGuard can achieve fast misuse detection with very low false positives and negatives while having negligible negative impact on legitimate data transmissions.

IX. Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments and helpful advice. This work was supported in part by the US National Science Foundation under grants CNS-1421999, CNS-1320906, CNS-1117462, and CNS-1422301. It was also supported by the Natural Science Foundation of China (NSFC) under Grants 61202140 and 61328208, by the Program for New Century Excellent Talents in University under Grant NCET-13-0548, and by the Innovation Foundation of the Chinese Academy of Sciences under Grand CXJJ-14-S132

References

[1] L. Yang, Z. Zhang, B. Zhao, C. Kruegel, and H. Zheng, “Enforcing dynamic spectrum access with spectrum permits,” in *ACM MobiHoc'12*, Hilton Head Island, SC, June 2012, pp. 195-204.

[2] V. Brik, V. Shrivastava, A. Mishra, and S. Banerjee, “Towards an architecture for efficient spectrum slicing,” in *HotMobile'07*, Tucson, AZ, Feb. 2007, pp. 61-69.

[3] W. Xu, P. Kamat, and W. Trappe, “TRIESTE: A trusted radio infrastructure for enforcing spectrum etiquettes,” in *IEEE Workshop on SDR Networks*, Reston, VA, Sept. 2006, pp. 101-109.

[4] G. Denker, E. Elenius, R. Senanayake, M. Stehr, and D. Wilkins, “A policy engine for spectrum sharing,” in *IEEE DySPAN'07*, Dublin, Ireland, Apr. 2007, pp. 55-65.

[5] S. Liu, L. Greenstein, Y. Chen, and W. Trappe, “ALDO: An anomaly detection framework for dynamic spectrum access networks,” in *IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009, pp. 675-683.

[6] I. Akyildiz, B. Lo, and R. Balakrishnan, “Cooperative spectrum sensing in cognitive radio networks: A survey,” *Physical Communication*, vol. 4, no. 1, pp. 40-62, Mar. 2011.

[7] R. Chen, J. Park, and K. Bian, “Robust distributed spectrum sensing in cognitive radio networks,” in *INFOCOM'08*, Apr. 2008, pp. 1876-1884.

[8] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. Hou, “Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks,” in *INFOCOM'12*, Orlando, FL, Mar. 2012.

[9] R. Zhang, *et al.*, “Secure crowdsourcing-based cooperative spectrum sensing,” in *INFOCOM'13*, Turin, Italy, Apr. 2013, pp. 2526-2534.

[10] R. Chen, J. Park, and J. Reed, “Defense against primary user emulation attacks in cognitive radio networks,” *IEEE JSAC*, vol.26, no.1, pp. 25-37, Jan. 2008.

[11] Y. Liu, P. Ning, and H. Dai, “Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures,” in *S&P'10*, Oakland, CA, May 2010, pp. 286-301.

[12] X. Tan, K. Borle, W. Du, and B. Chen, “Cryptographic link signatures for spectrum usage authentication in cognitive radio,” in *WiSec'11*, Hamburg, Germany, June 2011, pp. 79-90.

[13] K. Borle, B. Chen, and W. Du, “A physical layer authentication scheme for countering primary user emulation attack,” in *ICASSP'13*, Vancouver, Canada, May 2013, pp. 2935-2939.

[14] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, “Secret agent radio: Covert communication through dirty constellations,” in *IH'12*, Berkeley, CA, May 2012, pp. 160-175.

[15] V. Kumar, J. M. Park, T. C. Clancy, and B. Kaigui, “PHY-layer authentication by introducing controlled inter symbol interference,” in *CNS'13*, Washington, D.C., Oct. 2013, pp. 10-18.

[16] “Cisco visual networking index global mobile data traffic forecast update 2012-2017.” [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

[17] O. Fatemieh, R. Chandra, and C. Gunter, “Secure collaborative sensing for crowd sourcing spectrum data in white space networks,” in *DySPAN'10*, Singapore, Apr. 2010, pp. 1-12.

[18] A. Min, X. Zhang, and K. Shin, “Detection of small-scale primary users in cognitive radio networks,” *IEEE JSAC*, vol. 29, no. 2, pp. 349-361, Feb. 2011.

[19] “SHA-1,” http://en.wikipedia.org/wiki/SHA-1, [Online].

[20] T. Dierks and E. Rescorla, “The transport layer security (TLS) protocol,” RFC 4346, Apr. 2006.

[21] A. Goldsmith, “Wireless communications,” pp. 172-197, 2005.

[22] M. Morelli and U. Mengali, “A comparison of pilot-aided channel estimation methods for ofdm systems,” *Signal Processing, IEEE Transactions on*, vol. 49, no. 12, pp. 3065-3073, Dec. 2001.

[23] A. Wiesel, J. Goldberg, and H. Messer-Yaron, “Snr estimation in time-varying fading channels,” *Communications, IEEE Transactions on*, vol. 54, no. 5, pp. 841-848, May 2006.

[24] “How to: Define Minimum SNR Values for Signal Coverage,” http://www.wi-fiplanet.com/tutorials/article.php/3743986, [Online].