SpecGuard: Spectrum Misuse Detection in Dynamic Spectrum Access Systems

Xiaocong Jin, *Member, IEEE,* Jingchao Sun, *Member, IEEE,* Rui Zhang, *Member, IEEE,* Yanchao Zhang, *Senior Member, IEEE,* Chi Zhang, *Member, IEEE*

Abstract—Dynamic spectrum access (DSA) is the key to solving worldwide spectrum shortage. The open wireless medium subjects DSA systems to unauthorized spectrum use by illegitimate users. Secondary-user authentication is thus critical to ensure the proper operations of DSA systems. 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. The first scheme relies on a higher transmission power to embed the spectrum permit. To alleviate the assumptions on the additional transmission power, the second scheme is proposed with a limited negative impact on the normal data transmission. The third scheme takes a different approach by adopting a novel constellation design and exploiting the trust between the transmitter and the receiver. Crowdsourced spectrum misuse detection eliminates the need for the deployment of dedicated sensors and thus greatly reduces the deployment and maintenance cost. Detailed theoretical analyses, MATLAB simulations, and USRP experiments confirm that our schemes can achieve correct, low-intrusive, and fast spectrum misuse detection.

Index Terms—Dynamic spectrum access, spectrum misuse, security.

1 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 [2]. 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 involve 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. Authentication via the application layer is not preferred because applications are generally unaware of the underlying channels being used for data communications [3].
- *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 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 mis-

- *X. Jin is with Google LLC, Mountain View, California 94043 USA. This work was completed when he was with Arizona State University as a Ph.D. student.*
- *J. Sun is with Verizon, Sunnyvale, California 94089 USA.*
- *Y. Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA.*
- *R. Zhang is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716 USA.*
- *C. Zhang is with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027 China.*

Manuscript received xx, xx; revised xx, xx. An earlier version of this paper appeared in [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

use in DSA systems. The first approach assumes a tamper-proof transceiver to prevent unauthorized spectrum access [4], [5], [6], 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 [7]. A more recent method, Gelato [2], 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 it may be prohibitive for the spectrum owner to deploy adequate dedicated misuse detectors in a large geographic region to ensure sufficient coverage, 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 [8], 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.

The rest of the paper is organized as follows. In Section 2, we present the related work. Section 3 models the system and the adversary. This is followed by an overview of the SpecGuard in Section 4. Three schemes, different in spectrum permit transmission and detection, are detailed in Section 5. We conduct theoretical analyses in Section 6. In Section 7, practical implementation issues with USRP are discussed. A thorough performance evaluation is conducted in Section 8. Finally, Section 9 concludes our work.

2 Related Work

The work in [4], [5], [6] proposes to equip secondary users with tamper-resistant wireless transceivers to enforce spectrum policies and prevent them from illegitimately using the spectrum. Such tamper-resistant devices are expensive to build and subject to capable attacks. In contrast, SafeDSA does not require any tamper-resistant wireless transceiver on secondary users.

There has been significant effort on secure and/or privacy-preserving cooperative spectrum sensing. The work in [9], [10], [11], [12] aims to mitigate false sensing reports about the presence/absence of primary spectrum users. The work in [13], [14] studies location privacy issues when the physical locations and sensing reports from crowdsourcing users are correlated. In addition, the work in [15] identifies a new location inference attack in database-driven DSA systems by exploiting the channel usage history of secondary users. The work in [16], [17] incorporate differential privacy into crowdsourced spectrum sensing to protect location privacy for mobile participants. This body of work is orthogonal to SpecGuard.

Another line of work [18], [19], [20] 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 [20], 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 [21]. 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 [20], [21] 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 [22] 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 [22] 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.

The schemes in [2], [23], [24], [25], [26] are all PHY-based approaches for authenticating secondary users and most germane to SpecGuard. Gelato [2], [23] targets OFDM, the prevailing technology for wireless communications. In Gelato, every secondary user embeds a spectrum permit

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

by intentionally creating cyclostationary features in ODFM symbols. Gelato requires the repetition of multiple subcarriers to generate the desired and detectable cyclostationary feature, thus decreasing the data throughput. Cyclostationary feature detection also has high computational complexity and extremely long sensing time [8]. Kumar *et al.* proposed a PHY-layer authentication scheme [24] by introducing controlled inter symbol interference to identify rogue transmitters in DSA systems. This scheme, however, has a high error rate for the authentication bits and also negative impact on normal data transmissions. Moreover, no practical experiments were reported in [24]. FEAT [25] embeds the spectrum permit into the transmitted waveform by inserting an intentional frequency offset, and the verifier can decode the spectrum permit via frequency offset estimation with little knowledge about the transmission parameters. It is, however, computationally intensive to estimate the transmission parameters and thus the frequency offset. More recently, SafeDSA [26] proposes to embed the spectrum permit bits by dynamically changing the cyclic prefix length of each OFDM frame. The scheme is proved to work robustly in both AWGN and multipath Rayleigh fading channels. However, this scheme assumes limited multi-path delay spread, which might only be applicable to limited scenarios.

3 System and Adversary Models

3.1 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 [27] projects that the number of mobile-connected devices will hit 10 billion in 2019, 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 [28], [29]. 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.

The SpecGuard operator needs the locations of all misuse detectors to choose some for every instance of spectrum misuse detection. If misuse detectors are wary of location privacy, we can resort to a third-party trust broker as in [11]. The location privacy of misuse detectors can be well preserved so long as the SpecGuard operator and the trust broker do not collude. Given the focus of this paper, we refer interested readers to [11] for more details.

3.2 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 [4], [5], [6] 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.

4 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).

4.1 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 [30] 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 are secured using traditional mechanisms such as TLS [31].

4.2 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 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

4.3 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.

5 Spectrum-Permit Transmission and Detection

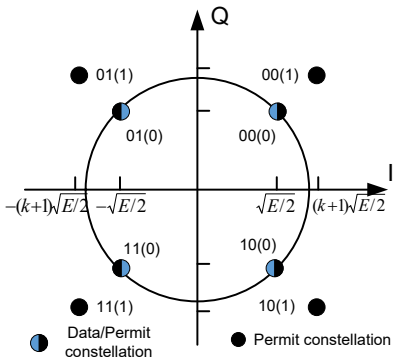
In this section, we detail how spectrum permits are transmitted and detected. There are two critical design constraints. First, the negative impact on the receiver's signal receptions should be very small. Second, misuse detectors are resource-constrained mobile users and should not perform expensive operations such as cyclostationary-feature detection. We propose to embed a spectrum permit through proper power control in the modulation phase and detect it in the demodulation phase of misuse detectors. In what follows, we first outline some background of QPSK and then present three schemes for transmitting and detecting spectrum permits.

5.1 QPSK Background

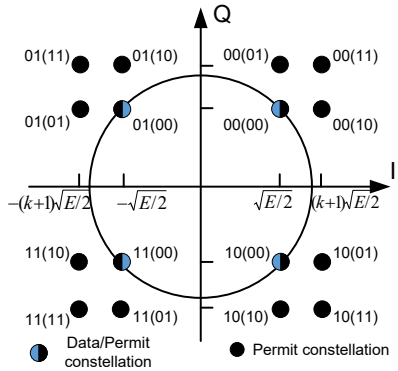
We assume QPSK as the physical-layer modulation scheme to ease the presentation, though our schemes can easily support general QAM. QPSK is a primitive modulation scheme in many applications and standards such as IEEE 802.11b, IEEE 802.11g and Bluetooth 2. It changes the phases of in-phase (I) and quadrature (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.

5.2 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 forward error correction (FEC) encoding to the spectrum permit.



(a) $M = 2$.



(b) $M = 4$.

Fig. 1: Constellation for Scheme 1.

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 to be discussed shortly.

5.2.1 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 6. 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,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

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 "0101011010101111111" 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 [32]. 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 [32]. 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.

5.2.2 Permit detection and verification

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. We can support data packets of variable lengths. A detection session may involve one or multiple packets, which depends on the lengths of data packets and spectrum permits.

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 8.

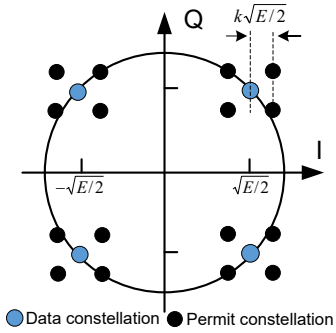


Fig. 2: Constellation for Scheme 2.

According to Section 4.3, each detector verifies a candidate spectrum permit constructed from consecutive permit bits detected from physical-layer symbols. It reports spectrum misuse to the operator whenever a valid spectrum permit is not detected in a detection session.

Permit transmission and detection in Scheme 1 are completely transparent to the receiver by assuming that phase tracking can be perfectly achieved. 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 to the receiver. This aspect will be further analyzed in Section 6. If the assumption about the perfect phase tracking does not hold, then we have to resort to the method proposed in Section 7 to correct any phase deviation introduced during the spectrum-permit embedding process.

5.2.3 Transmission parameters

Scheme 1 involves four key transmission parameters: E, k, m, and M. The transmitter can easily determine E by estimating the SNR [33], [34]. According to our analytical results in Section 6, 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. Note that the transmitter is naturally motivated to upload these parameters, as otherwise misuse detectors will report spectrum misuse when valid spectrum permits cannot be detected. The associated communication overhead is negligible if the data session lasts sufficiently long.

5.3 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

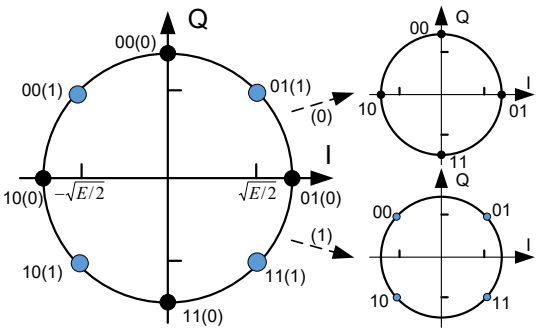


Fig. 3: Constellation for Scheme 3.

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 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 6. Other operations of Scheme 2 are similar to those of Scheme 1.

5.4 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.

5.4.1 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 least 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 bit 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 6. In addition, if a packet is not large 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.

5.4.2 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

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.

6 Theoretical Analysis

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

6.1 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 8.2, in practice, these assumptions may not be valid due to various channel conditions and effects. Based on the above assumptions, we have the following results.

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

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

1. The $\mathbf{erfc}()$ is the complementary error function, defined as $1 - \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\mathbf{d}t$.

and the permit BER for $M = 4$ is

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

Proof: From [32], the symbol error rate (SER) is approximately $P_s \approx \frac{W_{d_{\min}}}{2}\mathbf{erfc}(\frac{d_{\min}}{2\sqrt{N_0}})$, where $d_{\min}$ refers to the minimum distance between any two constellation points, and $W_{d_{\min}}$ records the number of neighbors at this distance. When M is 2, $d_{\min}$ equals $k\sqrt{E}$ and $W_{d_{\min}}$ equals one. So we obtain Eq. (1). When M is 4, $d_{\min}$ equals $k\sqrt{E/2}$, and $W_{d_{\min}}$ equals 2. With Gray coding, we can estimate the BER as half of the SER in Eq. (2). □

Theorem 2. The permit BER for Scheme 2 is

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

Proof: The minimum distance between the permit-constellation points is now $2k\sqrt{E/2}$. Eq. (3) can thus be derived similarly to Eq. (1). □

Theorem 3. The permit BER for Scheme 3 is

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

Proof: Since the minimum distance between permit-constellation points becomes $2\sin(\pi/8)\sqrt{E}$, we can similarly obtain Eq. (4) as Eq. (1). □

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 5.2.2, 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 8. Since the soft-decision always outperforms the hard-decision when the bits are repeated, the PER for the latter can be used as an upper bound of the PER for the former.

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

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

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

Proof: A hard decision is correct about a single bit only if there are at least $\lceil m/2\rceil$ repeated bits correctly received by the detector. So we can easily obtain Eq. (5). □

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 8.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

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.

6.2 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 8 to show that SpecGuard can achieve a small T.

6.3 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

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

and lower-bounded by

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

Proof: According to [32], the original BER for the QPSK modulation is as in Eq. (6). The upper bound of the data BER (worst case) is achieved when the permit bits are all 0s so that the absolute amplitude of all data symbols are still $\sqrt{E/2}$. We thus have Eq. (6). In contrast, the data BER can be minimized when the permit bits are all 1s so that the absolute amplitude of all data symbols is $(k+1)\sqrt{E/2}$. So we have Eq. (7). □

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

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

and lower-bounded by

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

Proof: When the amplitudes for both components are always decreased, the performance is the worst. Thus, the upper bound can be derived assuming the mutual distance between the QPSK constellation points is $2(1-k)\sqrt{E/2}$. Based on the nearest neighbor approximation, Eq. (8) is obtained. Correspondingly, the lower bound is achieved when the amplitudes for both components are always increased. In this case, the mutual distance between the QPSK constellation points is $2(1+k)\sqrt{E/2}$. Hence, Eq. (9) is derived. □

Given the decoding process in Section 5.4.2, the data BER of the receiver in Scheme 3 is the same as in the original QPSK constellation, i.e., $\mathbf{erfc}(\sqrt{\gamma/2})/2$.

6.4 Computation and Communication Overhead

The overhead in terms of computation and communication brought by SpecGuard is very limited.

We first analyze the computation overhead. In Scheme 1 and Scheme 2, the transmission power is adjusted according to the current spectrum permit bits to embed. On the transmitter end, the additional complexity is only due to the calculation of the shift from the original constellation points. It accounts for $\mathcal{O}(n)$ computation complexity, where n is the number of bits or symbols in the overall transmitted copies of spectrum permit. On the receiver end, it can either choose to completely ignore the embedded spectrum permit or fully recognize the phase deviations of the received samples from the standard constellation points to make corresponding corrections (to be detailed in Section 7). When the channel condition permits (i.e., SNR is high enough) and k is not too large, the receiver can choose the first strategy to simplify the implementation and thus achieve zero additional computational overhead. If the receiver chooses the second strategy to recover the small phase deviations, the additional computation overhead of $\mathcal{O}(n)$ complexity is added by first locating the correct point in the modified constellation and then performing phase recovery. In Scheme 3, a specially designed 8-PSK constellation is adopted by both the transmitter and the receiver. Thus, the additional computation complexity for the coding is $\mathcal{O}(n)$. In addition, both the transmitter and the receiver need to compute the γ spectrum permits for each time slot and thus involve a computation complexity of $\mathcal{O}(\gamma)$. Since γ is usually much smaller than n, the overall computation complexity is thus $\mathcal{O}(n)$. The detectors in all the schemes thus incur the computation overhead of $\mathcal{O}(n)$.

As for the communication overhead, common in all three schemes, the legitimate users need to purchase spectrum permit from the operator by specifying the desired channel

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

index, the geographic cell index and the time duration. This communication with the operator involves a limited communication overhead. In both Scheme 1 and Scheme 2, during the initialization phase, the operator needs to send n_0 to each misuse detector and n_γ to the legitimate transmitter. When the legitimate transmitter begins transmission, there is no additional communication overhead since the spectrum permit is embedded into the signal and thus does not cost additional samples. In Scheme 3, in addition to the communication overhead involved in Scheme 1 and Scheme 2, the spectrum permit needs to be shared with the targeted receiver. Hence, in this case, the legitimate transmitter needs to send n_γ to the targeted receiver as well.

6.5 From Unicast to Multicast

So far, we only focused on a single transmitter-receiver pair, i.e., the unicast case. In practice, it is also common that multicast transmissions are conducted, where one transmitter sends packets to multiple receivers. We investigate the feasibility of applying SpecGuard in this case and examine the additional overhead involved.

In Scheme 1 and Scheme 2, each receiver behaves individually, and there are no additional operations required for either the transmitter or the operator because the spectrum-permit transmission can be transparent to the receivers. Thus, Scheme 1 and Scheme 2 can be easily extended to accommodate the multicast scenario.

In Scheme 3, however, the transmitter and the receivers need to have bidirectional trust relationship. As the number of multicast receivers grows, it could be challenging to ensure that the transmitter can trust every receiver. Certain receivers in this case might become malicious by sharing certain spectrum permits or simply n_γ to other attackers for misusing the spectrum. Even though the transmitter can realize that the spectrum permit has been leaked, it is difficult to identify who is(are) the leaker(s) disclosing the spectrum permits. This could severely affect SpecGuard's operations. A simple solution could be that the transmitter hides the fact that he is conducting the multicast communication. If every receiver only knows that he is the targeted receiver but no one else, it is likely that he will not misbehave by leaking the spectrum permits. However, sometimes the receiver could simply identify that the communication is a multicast session even though he is not informed of it. For example, by decoding the contents of the packets, the receiver found that the intended receivers are a group of receivers who share certain common properties. In this case, the receiver could also possibly misbehave. A more technical solution will be included in future work. Additionally, as the number of receivers increase, the transmitter in Scheme 3 needs to proportionally send n_γ to each receiver, resulting in higher communication overhead. Fortunately, due to the limited communication range, the additional communication overhead can be very limited.

6.6 Benefits and Challenges in Crowdsourcing

Our proposed method relies on crowdsourcing. The unique merit of outsourcing spectrum misuse detection to mobile users is that it avoids the prohibitive cost of deploying a network of dedicated detectors needed by [2]. In addition, it can also ensure sufficient detector coverage, especially in metropolitan areas where the spectrum demands are the highest. Some crowdsourcing systems such as PocketSniffer [35] have already demonstrated the feasibility of crowdsourcing for wireless network access scenarios.

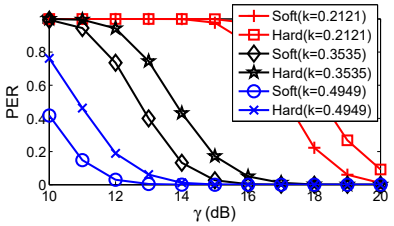


Fig. 4: Soft decision vs. hard decision.

Crowdsourcing-based spectrum misuse detection also faces challenges. A notable one is that the crowdsourcing detectors could potentially lie about the detection results [28], which may incur false positives and jeopardize legitimate transmission. For example, malicious detectors could report transmissions from legitimate transmitters as illegitimate, which may prompt the SpecGuard operator to carry out subsequent identification and punitive actions against legitimate transmitters. The impact of such attack could be even more severe if the malicious crowdsourcing detectors are the majority. We resort to the existing works such as [11], [12] for solutions in this regard.

7 Implementation Issues

We prototyped SpecGuard using USRP N210 with GNU Radio. This design is platform independent so that it can be ported to other hardware platforms as well. Moreover, since many components are not optimized, the performance our prototype achieved might not be the best performance that can be achieved using an advanced commercial platform. Below, some hardware implementation issues are briefly discussed.

Phase Ambiguity. QPSK suffers from phase ambiguity, a condition due to the nonlinear operation performed on the signal for carrier regeneration. The phase lock loop (PLL) could lock into a wrong phase, as a result of which, all the decoded data could be wrong. As discussed in Section 5.2 and Section 5.3, we adopt a special coding strategy to minimize the negative impact of this issue on the permit decoding when two permit bits are embedded with one data symbol. For Scheme 1 ($M = 2$), which embeds one permit bit per data symbol, the phase ambiguity will not affect permit decoding since the permit bit can be purely decided by the amplitude of the received symbol. For Scheme 3, however, although still only one permit bit is embedded with one data symbol, the permit bit can only be decided by the phase of the received symbol. Therefore, as detailed in Section 5.4, the original decoded bit sequence along with the bit-wise reverted one is used for the permit verification to mitigate this issue.

Automatic Gain Control. Automatic gain control (AGC) is widely adopted in receivers to enable dynamic adjustment of receiving gain. For QPSK and QAM modulation, the specified level usually corresponds to the mean power

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

of all the constellation points. In SpecGuard, since Scheme 1 and Scheme 2 modify the original constellation points to embed the permit information, the mean power of all the constellation points also changes. Hence, it is imperative to adjust the target gain level accordingly. Specifically, the target power level is $(1 + k + k^2/2)E$ in Scheme 1 and $(1 + k^2)E$ in Scheme 2. Since we only modify the phases of constellation points in Scheme 3, the target power level in AGC remains as the original level.

Phase Tracking. In a practical design, due to existing channel effects, the phase of the received signal might be changed from that of the signal sent. Therefore, phase recovery and phase tracking are vital in correct signal decoding. Costas loop is usually adopted as the component to enable phase and frequency synchronization. The essential idea is that the Costas loop first finds the error of the incoming signal symbol compared with its nearest constellation point, and then the frequency and phase of the numerically controlled oscillator (NCO) are updated according to this error. In Schemes 1 and 2, by changing the amplitude of the I and Q components of the signal, phase deviation between the added constellation point and the original constellation point might be introduced according to the value of M. Therefore, it is required that the detector should first find the correct quadrant (corresponding to data bits) and then decide the correct permit bits. In this way, the phase tracking can be done correctly. Otherwise, the whole decoding process could be wrong due to incorrect phase tracking.

8 Performance Evaluation

In this section, we evaluate SpecGuard using MATLAB simulations and USRP experiments. We also compare SpecGuard with [20] despite their different application scenarios. In addition, we apply SpecGuard to OFDM and compare the performance with two related schemes, FEAT [25] and SafeDSA [26]. Lastly, we evaluate the effectiveness of crowdsourcing within the proposed framework in MATLAB simulations.

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 of 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 6 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 [36], the channel SNR in [10,15), [15, 25), and [25, 40) indicates very poor, poor, and very good wireless channels, respectively. Finally, the two cases in Scheme 1 ($M = 2$ or 4) are differentiated by Scheme 1.1 and Scheme 1.2 whenever necessary.

8.1 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. 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 of using repetition codes. In general, the larger m, the lower PER, and vice versa.

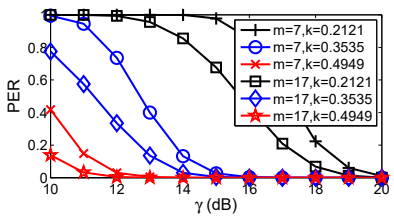
We also evaluated the PER for Scheme 3 in MATLAB. When γ equals $11|12|13|14|15|16|17|18$ dB, the PER is $1.00|0.99|0.92|0.66|0.31|0.07|0.02|0.00$. This result highlights the superior permit detection performance of Scheme 3 in contrast to Schemes 1 and 2. One may note that all our schemes have very high PERs when $\gamma \in [10, 15]$ dB. As mentioned above, $\gamma \in [10, 15]$ corresponds to very poor wireless channels over which normal data transmissions are unlikely to occur [36]. 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 6.1), 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 $\alpha = 5$.

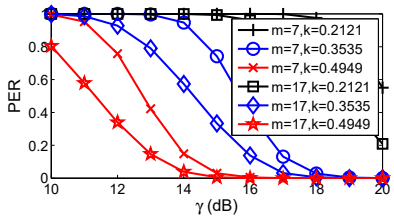
Moreover, we associate the results in Fig. 5 with the analysis in Section 6.2 to evaluate the fast property of SpecGuard. Here we let the data-transmission rate $R = 2$ Mb/s and the repetition parameter $m = 17$. Fig. 7 shows the impact of l (data-payload length) on the average permit detection time 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 evaluate 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-

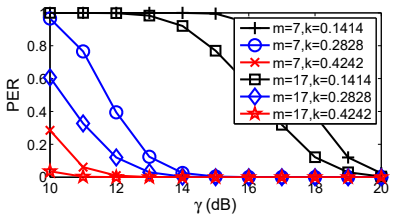
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing



(a) Scheme 1.1



(b) Scheme 1.2



(c) Scheme 2

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

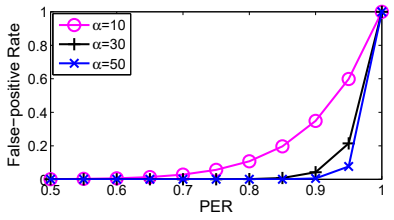


Fig. 6: False-positive rate.

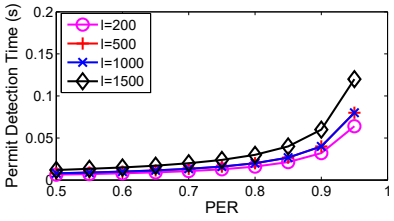


Fig. 7: Average permit detection time.

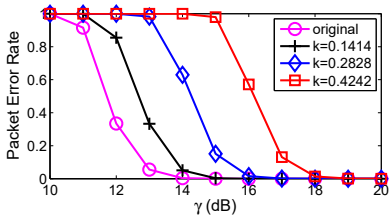


Fig. 8: Data error rate for Scheme 2.

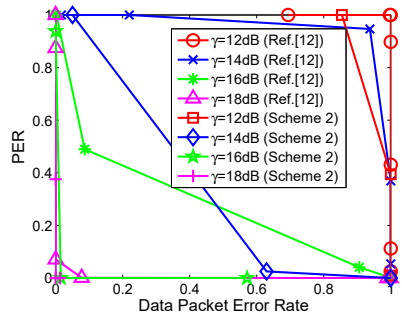
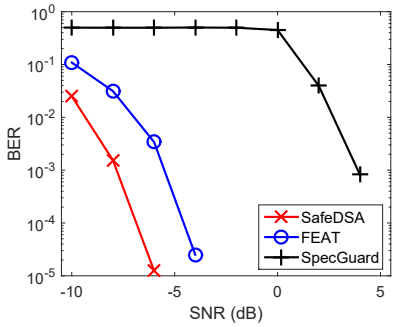
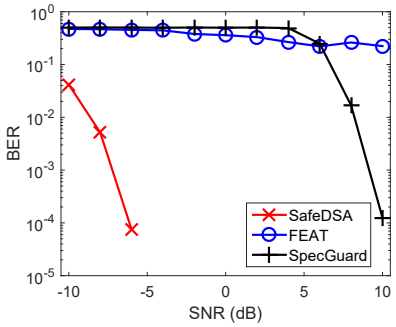


Fig. 9: Comparison between Scheme 2 and [20].



(a) AWGN channel.



(b) Multipath Rayleigh fading channel.

Fig. 10: Permit BER comparison for different channels.

constellation points (cf. Fig. 2). Scheme 2 still works well for high SNRs.

Table. 1 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.

We jointly compare the permit and data decoding performance of Scheme 2 with the work in [20] in Fig. 9. In the comparison, we fixed $m = 7$ and varied the value of k. For [20], the shifted angle was changed from 0.1 to 0.7 rad. Generally, the closer the curves to the origin, the lower

TABLE 1: 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%

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 [20].

In addition, we implement SpecGuard in an OFDM framework and compare it with two other OFDM-based schemes, FEAT [25] and SafeDSA [26]. It should be noted that different from FEAT and SafeDSA, SpecGuard is independent of the modulation schemes. Hence, for other non-OFDM systems such as any single-carrier systems, SpecGuard can be adopted as well and thus has wider applicability. Since both FEAT and SafeDSA embed one spectrum permit bit per OFDM frame, we adopt the same strategy for SpecGuard (i.e., we let M be 2). This also means

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing

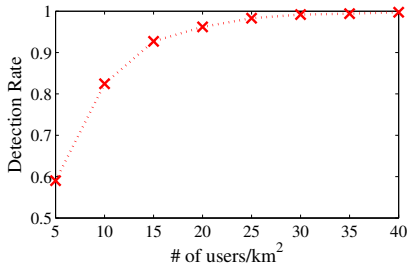


Fig. 11: True positive rate with crowdsourcing.

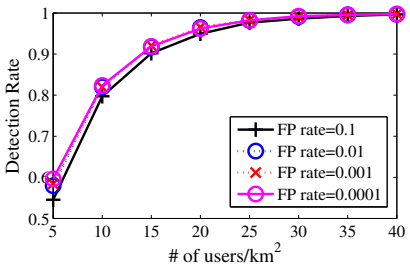


Fig. 12: True negative rate with crowdsourcing.

that the value of m will be larger than that used in all the previous simulations.

The detailed configurations are as below. In FEAT, the sampling frequency is set as 1 MHz, and the maximum positive frequency offset for embedding the spectrum permit is 5 kHz. Recall that Scheme 1 in SpecGuard has a generally higher additional transmission power requirement and that Scheme 3 requires additional trust relationship between the sender and the receiver. We thus employ Scheme 2 for a fair comparison, with k set as 0.14 (the corresponding additional power is 2% according to Table. 1). We let the FFT size in OFDM be 64, the cyclic prefix length be 16, and each frame consists of 25 OFDM symbols. In addition, we assume that 52 out of the 64 subcarriers are used for data transmission while others are simply pilot subcarriers. Hence, m is equal to 1,300.

We conduct the performance comparisons for two different channel types: AWGN and multipath Rayleigh fading channels. Fig. 10 shows the evaluation results. Generally speaking, since the simulated SNR values are so small, we can safely say that SpecGuard can perform reliably for both channel conditions. Specifically, the BER curve of SpecGuard descends very quickly when SNR is over 0 dB in the AWGN channel and when SNR is over 5 dB in the multipath Rayleigh fading channel. Also we note that when SNR is close to or below 0 dB, SpecGuard's BER curve is simply horizontal, which is due to the fundamental performance limitation of the PSK modulation itself, while other schemes does not suffer this disadvantage. On the other hand, since the purpose of the secondary transmission is to achieve a desired data throughput, which is modulated using PSK or other fundamental modulations, it is also not necessary to get authenticated by the detector if the normal data transmission fails constantly. Thus, as long as SpecGuard can perform reliably for both channels with a generally reasonable SNR (eg., above 14 dB [37]), SpecGuard suffices spectrum permit detection. Although SafeDSA achieves very low BERs at the left end of the simulated SNR regions, it requires the receiver to decode the spectrum permits as well, thus introducing additional computation overhead. In addition, the accurate decoding of the spectrum permits is the primitive step for decoding the data bits for the receiver in SafeDSA, while for all the schemes in SpecGuard, the spectrum permits are not required to be decoded by the receivers so that there is a lower performance requirement for the spectrum permit detection. On the other hand, FEAT delivers a reliable spectrum permit detection performance in the AWGN channel but it fails in the multipath Rayleigh fading channel. The root case of this failure is that FEAT requires enough samples to perform accurate parameter estimation. If the number of received samples is not enough, the estimation fails and thus generates false detection results. Clearly, SpecGuard is not limited to this bottleneck and can be very flexible with the number of repetitions and other transmission parameters.

We also evaluate the effectiveness of crowdsourcing with SpecGuard. We simulate a square area of size 2km by 2km with one transmitter. The transmitter's transmission range as well as the detector's detection range are both 250m. For this simulation, based on the results obtained earlier, we assume that the false-positive (FP) rate ranges from 0.0001 to 0.1 and that the true-positive rate is zero. The locations of the transmitter and the detectors are uniformly distributed at random. The simulation results are obtained by averaging the results from 10,000 runs.

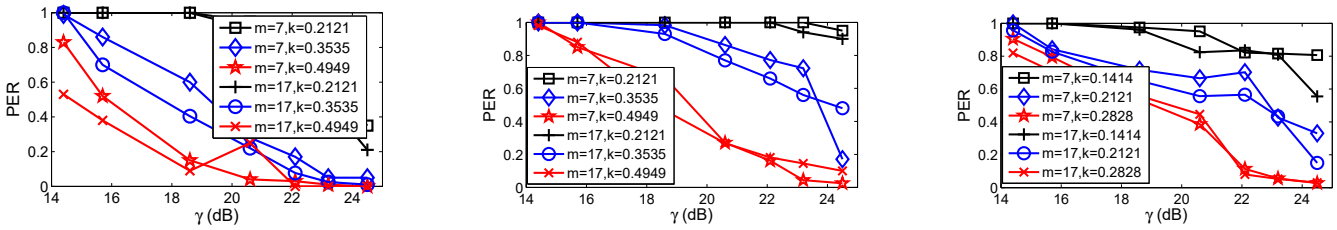
Fig. 11 shows the true positive rate, the probability of an illegitimate user being detected, with the density of crowdsourcing worker varying from 5 to 40 per km^2. When a detector is located within the transmitter's transmission range, the true positive rate is constantly one based on previous analysis. However, illegitimate transmissions will not be detected if no mobile detector is within the transmission range of the illegitimate transmitter. As we can see from the figure that with a low user density, e.g., 5 per km^2, the detection rate is only about 0.6. As the user density increases, the detection rate gradually increases. In our simulations, a user density of 40 per km^2 is sufficient to achieve a detection rate of close to one. Fig. 12 shows the true negative rate, the probability of an legitimate transmitter being detected as an legitimate one. There are four curves in the figure, corresponding to four values of the FP rates for a single detector. We can see that although a lower FP rate helps to improve the detection rate, the dominant factor is still the crowdsourcing detector's density.

8.2 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. 13 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2823314, IEEE Transactions on Mobile Computing



(a) Scheme 1.1 (b) Scheme 1.2 (c) Scheme 2

Fig. 13: PER performance using USRP.

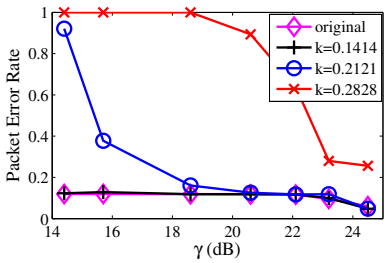


Fig. 14: Data-packet error rate for Scheme 2 using USRP.

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. 14, 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.

9 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.

Acknowledgments

This work was supported in part by the US National Science Foundation under grants CNS-1619251, CNS-1514381, CNS-1421999, CNS-1320906, CNS-1700032, CNS-1700039, CNS-1651954 (CAREER), and CNS-1718078. 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

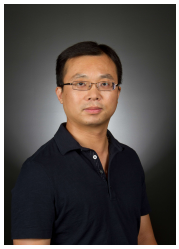
[1] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "SpecGuard: Spectrum misuse detection in dynamic spectrum access systems," in *INFOCOM'15*, Apr. 2015.
[2] L. Yang, Z. Zhang, B. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *MobiHoc'12*, June 2012.
[3] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, iss. 13, pp. 2127-2159, Sept. 2006.
[4] V. Brik, V. Shrivastava, A. Mishra, and S. Banerjee, "Towards an architecture for efficient spectrum slicing," in *HotMobile'07*, Feb. 2007.
[5] W. Xu, P. Kamat, and W. Trappe, "TRIESTE: A trusted radio infrastructure for enforcing spectrum etiquettes," in *IEEE Workshop on SDR Networks*, Sept. 2006.
[6] G. Denker, E. Elenius, R. Senanayake, M. Stehr, and D. Wilkins, "A policy engine for spectrum sharing," in *DySPAN'07*, Apr. 2007.
[7] S. Liu, L. Greenstein, Y. Chen, and W. Trappe, "ALDO: An anomaly detection framework for dynamic spectrum access networks," in *INFOCOM'09*, Apr. 2009.
[8] 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.
[9] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM'08*, Apr. 2008.
[10] 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*, Mar. 2012.
[11] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *INFOCOM'13*, Apr. 2013.
[12] Y. Hu and R. Zhang, "Secure crowdsourced Radio Environment Map construction," in *ICNP'17*, Oct. 2017.
[13] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *INFOCOM'12*, Mar. 2012.
[14] Z. Gao, H. Zhu, S. Li, and S. Du, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 106-112, Dec. 2012.

[15] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: attacks and countermeasures," in *INFOCOM'13*, Apr. 2013.
[16] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *INFOCOM'16*, Apr. 2016.
[17] X. Jin, R. Zhang, Y. Chen, T. Li, and Y. Zhang, "DPSense: Differentially private crowdsourced spectrum sensing," in *CCS'16*, Oct. 2016.
[18] R. Chen, J. Park, and J. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol.26, no.1, pp. 25-37, Jan. 2008.
[19] 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*, May 2010.
[20] X. Tan, K. Borle, W. Du, and B. Chen, "Cryptographic link signatures for spectrum usage authentication in cognitive radio," in *WiSec'11*, June 2011.
[21] K. Borle, B. Chen, and W. Du, "A physical layer authentication scheme for countering primary user emulation attack," in *ICASSP'13*, May 2013.
[22] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret agent radio: Covert communication through dirty constellations," in *IH'12*, May 2012.
[23] A. Nika, Z. Zhang, B. Zhao, and H. Zheng, "Toward practical spectrum permits," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 1, pp. 112-122, Mar. 2017.
[24] V. Kumar, J. Park, T. Clancy, and B. Kaigui, "PHY-layer authentication by introducing controlled inter symbol interference," in *CNS'13*, Oct. 2013.
[25] V. Kumar, J. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *CCS'14*, Nov. 2014.
[26] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "SafeDSA: Safeguard dynamic spectrum access against fake secondary users," in *CCS'15*, Oct. 2015.
[27] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021 white paper," Mar. 28, 2017.
[28] O. Fatemieh, R. Chandra, and C. Gunter, "Secure collaborative sensing for crowd sourcing spectrum data in white space networks," in *DySPAN'10*, Apr. 2010.
[29] A. Min, X. Zhang, and K. Shin, "Detection of small-scale primary users in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 349-361, Feb. 2011.
[30] "SHA-1," http://en.wikipedia.org/wiki/SHA-1.
[31] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol," RFC 4346, Apr. 2006.
[32] A. Goldsmith, "Wireless communications," pp. 172-197, 2005.
[33] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3065-3073, Dec. 2001.
[34] A. Wiesel, J. Goldberg, and H. Messer-Yaron, "SNR estimation in time-varying fading channels," *IEEE Trans. Commun*, vol. 54, no. 5, pp. 841-848, May 2006.
[35] J. Shi, Z. Guan, C. Qiao, T. Melodia, D. Koutsonikolas, G. Challen, "Crowdsourcing access network spectrum allocation using smartphones," in *HotNets-XIII*, Oct. 2014.
[36] "How to: Define minimum SNR values for signal coverage," http://www.wi-fiplanet.com/tutorials/article.php/3743986.
[37] "Design considerations," http://www.cisco.com/c/en/us/td/docs/wireless/technology/mesh/7-3/design/guide/Mesh/Mesh_chapter_011.pdf.

Xiaocong Jin received the B.E. in Information Engineering from Shanghai Jiao Tong University, China, in 2009, the M.S. in Information, Production, and Systems Engineering from Waseda University, Japan, in 2010, the M.S. in Signal and Information Processing from Shanghai Jiao Tong University, China, in 2012 and the Ph.D. in Electrical Engineering from Arizona State University, United States. He is now with Google LLC as a software engineer. His primary research interests are network and distributed system security and privacy, wireless networking, and mobile computing. He is a member of IEEE.

Jingchao Sun received the B.E. in Electronics and Information Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2008 and 2011, respectively, and the Ph.D. degree in Electrical Engineering from Arizona State University. He is now with Verizon. His primary research interests are network and distributed system security and privacy, wireless networking, and mobile computing. He is a member of IEEE.

Rui Zhang received the B.E. degree in communication engineering and M.E. degree in communication and information system from the Huazhong University of Science and Technology in 2001 and 2005, respectively, and the Ph.D. degree in electrical engineering from Arizona State University in 2013. He was an Assistant Professor with the Department of Electrical Engineering at the University of Hawaii from 2013 to 2016. He has been an Assistant Professor with the Department of Computer and Information Sciences at the University of Delaware since 2016. His current research interests include network and distributed system security, wireless networking, and mobile computing. He received the NSF CAREER Award in 2017 and is a member of IEEE.



Yanchao Zhang received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is an Associate Professor in School of Electrical, Computer and Energy Engineering at Arizona State University. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is an Editor of IEEE Transactions on Mobile Computing, IEEE Transactions on Vehicular Technology, and IEEE Wireless Communications. He was also a TPC Co-Chair of Communication and Information System Security Symposium, IEEE GLOBECOM 2010. He received the NSF CAREER Award in 2009 and is a senior member of IEEE.

Chi Zhang received the B.E. and M.E. in electrical and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. in electrical and computer engineering from the University of Florida, Gainesville, Florida, in 2011. He joined University of Science and Technology of China in September 2011 as an Associate Professor of School of Information Science and Technology. His research interests are in the areas of network protocol design, network performance analysis, and network security guarantee, particularly for wireless networks and social networks. He received the 7th IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award.