Security and Privacy in Dynamic Spectrum Access:

Challenges and Solutions

by

Xiaocong Jin

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved May 2017 by the
Graduate Supervisory Committee:

Yanchao Zhang, Chair
Junshan Zhang
Cihan Tepedelenlioglu
Lei Ying

ARIZONA STATE UNIVERSITY

August 2017

ABSTRACT

Dynamic spectrum access (DSA) has great potential to address worldwide spectrum shortage by enhancing spectrum efficiency. It allows unlicensed secondary users to access the under-utilized spectrum when the primary users are not transmitting. On the other hand, the open wireless medium subjects DSA systems to various security and privacy issues, which might hinder the practical deployment. This dissertation consists of two parts to discuss the potential challenges and solutions.

The first part consists of three chapters, with a focus on secondary-user authentication. Chapter One gives an overview of the challenges and existing solutions in spectrum-misuse detection. Chapter Two presents SpecGuard, the first crowdsourced spectrum-misuse detection framework for DSA systems. In SpecGuard, three novel schemes are proposed for embedding and detecting a spectrum permit at the physical layer. Chapter Three proposes SafeDSA, a novel PHY-based scheme utilizing temporal features for authenticating secondary users. In SafeDSA, the secondary user embeds his spectrum authorization into the cyclic prefix of each physical-layer symbol, which can be detected and authenticated by a verifier.

The second part also consists of three chapters, with a focus on crowdsourced spectrum sensing (CSS) with privacy consideration. CSS allows a spectrum sensing provider (SSP) to outsource the spectrum sensing to distributed mobile users. Without strong incentives and location-privacy protection in place, however, mobile users are reluctant to act as crowdsourcing workers for spectrum-sensing tasks. Chapter Four gives an overview of the challenges and existing solutions. Chapter Five presents PriCSS, where the SSP selects participants based on the exponential mechanism such that the participants' sensing cost, associated with their locations, are privacy-preserved. Chapter Six further proposes DPSense, a framework that allows the honest-but-curious SSP to select mobile users for executing spatiotemporal

spectrum-sensing tasks without violating the location privacy of mobile users. By collecting perturbed location traces with differential privacy guarantee from participants, the SSP assigns spectrum-sensing tasks to participants with the consideration of both spatial and temporal factors.

Through theoretical analysis and simulations, the efficacy and effectiveness of the proposed schemes are validated.

To my parents and my sister, who constantly inspire me to pursue my dreams.

To my beloved Jiaying, for her untiring support.

ACKNOWLEDGMENTS

I owe my gratitude to several people who have advised, supported, or inspired me during the course of the work.

First, I want to truly thank my advisor Dr. Yanchao Zhang, who through his immense patience and forbearance has shown me an exciting world of research. You have been a constant source and guide of knowledge for the past five years. Without your encouragement and help, I would not have completed this work. Thank you.

I would also like to acknowledge Dr. Junshan Zhang, Dr. Cihan Tepedelenlioglu, and Dr. Lei Ying, who have supported me in many different ways over these years. I greatly appreciate Dr. Zhang, Dr. Tepedelenlioglu and Dr. Ying for serving on my dissertation committee and providing me with guidance from time to time about my dissertation.

I have received the help and support from a great number of people, including, but not limited to, my colleagues Dr. Rui Zhang, Dr. Jinxue Zhang, Dr. Jingchao Sun, Yimin Chen, Tao Li, Dianqi Han, Dr. Junwei Zhang, Xin Yao, Dr. Mande Xie, and Dr. Xiaogang Qi, who I have closely worked with over the years.

My research work and the writing of this dissertation could not have been completed without the enormous support of my family. I thank my family members for being a constant source of support and encouragement. My deepest gratitude also goes to my friends who encourage and believe in me over the time. Lastly, I owe my gratitude to my beloved Jiaying, who is by my side and encouraging me all the time.

I also gratefully acknowledge the financial support I received from the National Science Foundation through grant CNS-0844972, CNS-1117462, and CNS-1320906.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

MISUSE DETECTION IN DYNAMIC SPECTRUM ACCESS

1.1 Introduction

Dynamic spectrum access (DSA) [5] 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 [6]. 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* [7, 8]. 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.

There have been recent efforts [6, 9, 10] to authenticate secondary users in DSA systems. Common in these schemes, a secondary user needs to embed into his physical-layer signals some cryptographic, unforgeable information, which we call a *spectrum permit* and serves as his credential for using a given spectrum band. A *verifier* authenticates the secondary user by detecting and verifying the spectrum permit. Verifiers can be dedicated entities of the spectrum owner [6] or mobile crowdsourcing users [8]. If a valid spectrum permit cannot be detected, verifiers can report to the spectrum owner which can take further actions such as triangulating the fake secondary user and involving law enforcement. Such PHY-based approaches are highly desirable in that they involve the physical layer only and will not interrupt the protocol operations at the data-link layer and above at the secondary user. These schemes use different features of the physical layer to embed the spectrum permit. In particular, Gelato [6] generates physical-layer cyclostationary features; P-DSA [9] adds controlled inter-symbol interference; FEAT [10] intentionally tunes the frequency offset. Although these schemes can all detect fake secondary users with very low false positives and negatives, Gelato and FEAT have high computational overhead.

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.

1.2 Related Work

This section reviews the prior work most related to our research issue.

The work in [11, 12, 13] 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.

The work in [14] uses a dedicated sensor network to perform spatially distributed power measurements for detecting illegitimate secondary users.

There has been some work [15, 16, 17] to construct a physical-layer covert channel which is not easily detectable by the adversary. Although our approaches introduced later also embeds information into physical-layer signals, they do not try to hide the embedded spectrum permit but instead aims to make it easily detected by any verifier who overhears the secondary user's transmission.

A large chunk of work (e.g., [18, 19, 20]) aims to mitigate fake sensing reports about the presence or absence of primary users. This line of research does not involve

secondary users and is orthogonal to our approaches. Authors in [21, 22] discuss location privacy issues found in spectrum sensing based on the strong correlations between the physical locations and the sensing values submitted. The work in [23] identifies a new attack where in database-driven DSA systems, SUs' locations can be inferred through their used channels. These work are all orthogonal to our work.

Another line of work [1, 24, 25] targets authenticating primary users in DSA systems. The attack under consideration is the primary user emulation attack in which unauthorized users pretend as the primary user to use the channel. By contrast, our approaches aims at authenticating secondary users who may or may not be authorized to use the channel.

As said, the schemes in [6, 9, 10] are all PHY-based approaches for authenticating secondary users and most germane to our approaches. As the seminal work, Gelato [6] targets OFDM, the prevailing technology for wireless communications. In Gelato, every secondary user embeds a spectrum permit by intentionally creating cyclostationary features in ODFM symbols. Gelato requires the repetition of multiple sub-carriers to generate the desired and detectable cyclostationary feature, thus decreases the data throughput. Cyclostationary feature detection also has high computational complexity and extremely long sensing time [26]. P-DSA [9] requires the transmitter to add controlled inter-symbol interference and the receiver to add maximum likelihood detection to extract the permit bits. However, the added inter-symbol interference still negatively impacts normal data transmission. FEAT [10] 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.

Chapter 2

SPECGUARD: A SPATIAL APPROACH FOR SPECTRUM MISUSE DETECTION

2.1 Overview

This chapter presents *SpecGuard* [8], the first crowdsourced spectrum misuse detection framework for DSA systems. Motivated by Gelato [6], SpecGuard requires a spectrum permit to be embedded into and detected from physical-layer signals. To address the 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 chapter is organized as follows. Section 2.2 models the system and the adversary. This is followed by an overview of the SpecGuard in Section 2.3. Three schemes, different in spectrum-permit transmission and detection, are detailed in Section 2.4. We conduct theoretical analyses in Section 2.5. In Section 2.6, practical implementation issues with USRP are discussed. A thorough performance evaluation is conducted in Section 2.7. Finally, Section 2.8 concludes our work.

2.2 System and Adversary Models

2.2.1 System Model

SpecGuard is in charge by an operator. The 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 devices and connections 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 [20]. 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 chapter, we refer interested readers to [20] for more details.

2.2.2 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 [11, 12, 13] 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.

2.3 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).

2.3.1 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 $h^{\lambda}(x)$ denote λ 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 $\kappa \geq 1$ slots. Upon receiving the spectrum-access request, the SpecGuard operator selects a random number n_{κ} of sufficient length (say, 160 bits), recursively computes $n_i = h(n_{i+1}), \forall i \in [0, \kappa - 1]$, and finally sends n_{κ} to the legitimate user who then recursively computes $\{n_0, \ldots, n_{\kappa-1}\}$. 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 [31].

2.3.2 Spectrum-Permit Transmission and Detection

The legitimate transmitter needs to keep transmitting the spectrum permit n_i in slot i ($\forall i \in [1, \kappa]$) 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 2.4.

2.3.3 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 $i \in [1, \kappa]$ 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.

2.4 Spectrum-Permit Transmission and Detection

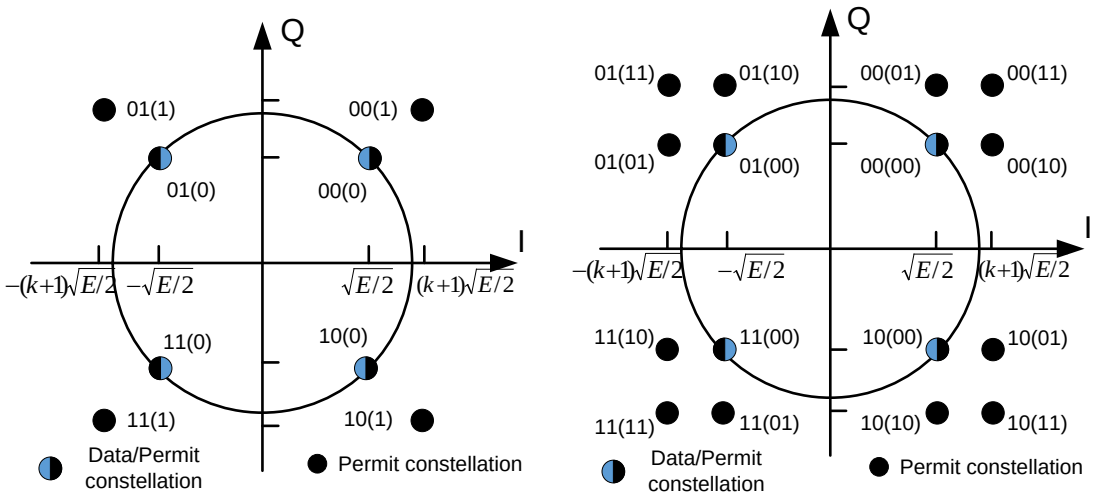
In this section, we detail how spectrum permits are transmitted and detected. There are two critical design constraints. First, the negative impact on the receiver's signal receptions should be very small. Second, misuse detectors are resource-constrained mobile users and should not perform expensive 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.

2.4.1 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.

2.4.2 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 FEC encoding to the spectrum permit. Although there are many FEC schemes



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

Figure 2.1: Constellation for Scheme 1.

available, we choose the repetition code for its simplicity. How the repetition code is implemented depends on the constellation design discussed shortly.

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 2.5. For clarity, we show the constellation graph for Scheme 1 in Fig. 2.1a, 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. 2.1a can be considered as a 2-QAM constellation for permit bits. An example for $M = 4$ is given in Fig. 2.1b, 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 "0101011010111111" 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. 2.1b. 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.

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. 2.1a. In the hard-decision strategy, the detector finds the constellation point in Fig. 2.1a 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 2.7.

According to Section 2.3.3, each misuse detector verifies a candidate spectrum permit constructed from consecutive permit bits detected from physical-layer symbols. It reports spectrum misuse to the SpecGuard operator whenever a valid spectrum permit is not detected in a detection session.

Permit transmission and detection in Scheme 1 are totally 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 transmission to the receiver. This aspect will be further analyzed in Section 2.5. If the assumption about the perfect phase tracking does not hold, then we have to resort to the method proposed in Section 2.6 to correct any phase deviation introduced during the spectrum-permit embedding process.

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 2.5, 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

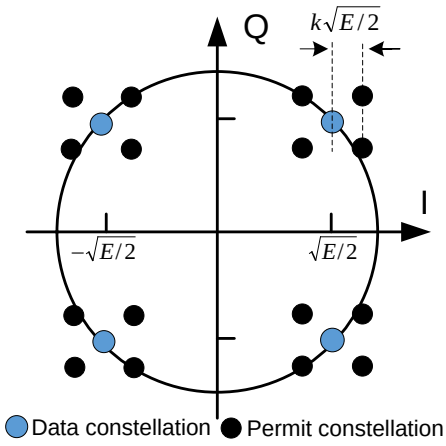


Figure 2.2: Constellation for Scheme 2.

when valid spectrum permits cannot be detected. The associated communication overhead is negligible if the data session lasts sufficiently long.

2.4.3 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 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 might already be 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. 2.2 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 2.5. Other operations of Scheme 2 are similar to those of Scheme 1.

2.4.4 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.

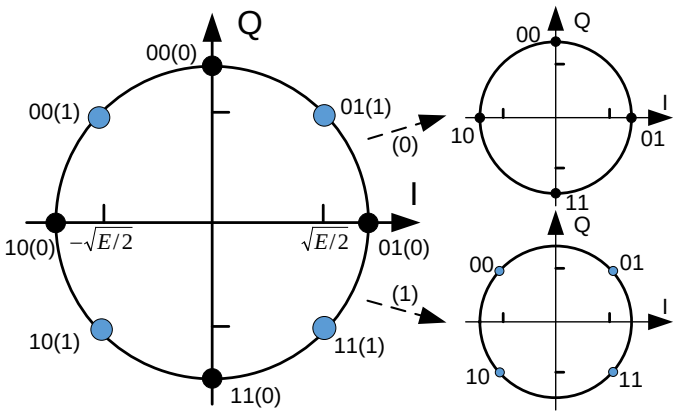


Figure 2.3: Constellation for Scheme 3.

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. 2.3. 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 2.5. 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.

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 2.3, 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.

2.5 Theoretical Analysis

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

2.5.1 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 2.7.2, 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

$$P_{b,1}^{M=2} \approx \mathbf{erfc}(k\sqrt{\gamma}/2)/2, \text{[1]} \tag{2.1}$$

and the permit BER for $M = 4$ is

$$P_{b,1}^{M=4} \approx \mathbf{erfc}(k\sqrt{\gamma/2}/2)/2. \tag{2.2}$$

Proof. According to [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

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

$W_{d_{\min}}$ 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. (2.1). 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. (2.2). □

Theorem 2. The permit BER for Scheme 2 is

$$P_{b,2} \approx \mathbf{erfc}(k\sqrt{\gamma/2})/2. \tag{2.3}$$

Proof. The minimum distance between the permit-constellation points is now $2k\sqrt{E/2}$. Eq. (2.3) can thus be derived similarly to Eq. (2.1). □

Theorem 3. The permit BER for Scheme 3 is

$$P_{b,3} \approx \mathbf{erfc}(\sqrt{\gamma}\sin(\pi/8)). \tag{2.4}$$

Proof. Since the minimum distance between permit-constellation points becomes $2\sin(\pi/8)\sqrt{E}$, we can similarly obtain Eq. (2.4) as Eq. (2.1). □

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 2.4.2, 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 2.7. 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

$$\begin{aligned} P_p =& 1 - (\binom{m}{\lceil m/2 \rceil}(1-P_b)^{\lceil m/2 \rceil} P_b^{m-\lceil m/2 \rceil} \\ &+ \binom{m}{\lceil m/2 \rceil + 1}(1-P_b)^{\lceil m/2 \rceil + 1} P_b^{m-\lceil m/2 \rceil - 1} \\ &+ ... + (1-P_b)^m)^L, \end{aligned} \tag{2.5}$$

where P_b is given in Eq. (2.1), Eq. (2.2), or Eq. (2.3).

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. (2.5). □

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 2.7.

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 transmission. 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.

2.5.2 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 n_x denote the number of data packets required to transmit a complete L-bit spectrum permit. We can easily compute n_x for different schemes: (1) $n_x = \lceil \frac{Lm}{4l} \rceil$ for Scheme 1 ($M = 2$); (2) $n_x = \lceil \frac{Lm}{8l} \rceil$ for Scheme 1 ($M = 4$) and Scheme 2; (3) $n_x = \lceil \frac{L}{4l} \rceil$ for Scheme 3. Given the PER P_p computed above, the average detection time for all the schemes is computed as $\widetilde{t} = \frac{8lx}{r(1-P_p)}$ seconds. Examples are given in Section 2.7 to show that SpecGuard can achieve a small $\widetilde{t}$.

2.5.3 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

$$\overline{\mathrm{BER}_{1,\mathrm{data}}} \approx \mathbf{erfc}(\sqrt{\gamma/2})/2, \tag{2.6}$$

and lower-bounded by

$$\underline{\mathrm{BER}_{1,\mathrm{data}}} \approx \mathbf{erfc}(\sqrt{(1+k^2)\gamma/2})/2. \tag{2.7}$$

Proof. According to [32], the original BER for the QPSK modulation is as in Eq. (2.6). 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. (2.6). 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. (2.7). □

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

$$\overline{\mathrm{BER}_{2,\mathrm{data}}} \approx \mathbf{erfc}((1-k)\sqrt{\gamma/2})/2, \tag{2.8}$$

and lower-bounded by

$$\underline{\mathrm{BER}_{2,\mathrm{data}}} \approx \mathbf{erfc}((1+k)\sqrt{\gamma/2})/2. \tag{2.9}$$

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. (2.8) 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. (2.9) is derived. □

Given the decoding process in Section 2.4.4, the data BER of the receiver in Scheme 3 is the same as in the original QPSK constellation, i.e., $\mathbf{erfc}(\sqrt{\gamma/2})/2$.

2.5.4 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 2.6). 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 κ spectrum permits for each time slot and thus involve a computation complexity of $\mathcal{O}(\kappa)$. Since κ 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 permits from the operator by specifying the desired

channel 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_κ 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_κ to the targeted receiver as well.

2.5.5 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_κ 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 sends n_κ to each receiver, resulting in a higher communication overhead. Fortunately, due to the limited communication range, the additional communication overhead can be very limited.

2.5.6 Benefits and Challenges in Crowdsourcing

Our method is based on crowdsourcing. The unique merit enabling SpecGuard to outsource spectrum-misuse detection to mobile-crowdsourcing workers is that it incurs very low additional computation and communication overhead as analyzed in Section. 2.5.4. Thus, dedicated detectors such as the ones proposed in [6] are not needed. One obvious benefit of crowdsourcing brought by SpecGuard is that sufficient detector coverage can be ensured where crowdsourcing workers actively participates. For physical regions without sufficient active crowdsourcing workers, the SpecGuard operator can dispatch some dedicated detectors as a complement. Managing such crowdsourcing platforms involves additional overhead for the operator, but it can significantly reduce the number of dedicated detectors and make the system easily scalable.

Along with all these benefits of crowdsourcing, challenges arise as well. One of the notable challenges [28] is that the crowdsourcing detectors could potentially lie about the detection results. This may cause severe interference to the normal transmission. For example, without the existence of any illegitimate spectrum users, a malicious detector could report that the current on-going suspicious transmission comes from an illegitimate user. This might lead the SpecGuard operator to conclude that the current transmitter does not possess a legitimate spectrum permit. The impact of this attack could be even more severe if the malicious crowdsourcing detectors are the majority. We resort to the existing work such as [20] for solutions in this regard. The essential idea is to jointly consider the instantaneous trustworthiness of mobile detectors in combination with their reputation scores. If combined with these schemes, the proposed method can effectively enable robust spectrum-permit detection based on crowdsourcing.

2.6 Implementation Issues

We prototyped SpecGuard using USRP N210 with GNU Radio. This design is platform independent so that it can be ported to other 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 2.4.2 and Section 2.4.3, 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

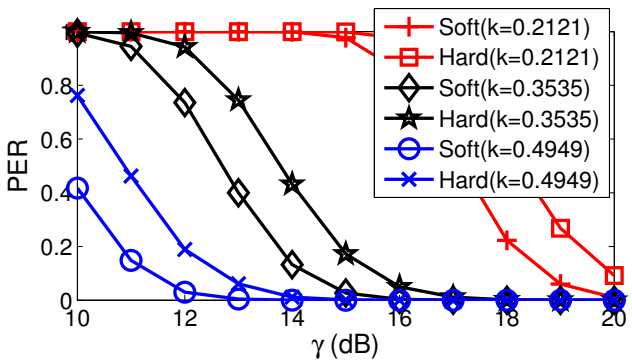


Figure 2.4: Soft Decision vs. Hard Decision.

($M = 2$), 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 2.4.4, the original decoded bit sequence along with the bit-wise reverted one is used for the permit verification to mitigate this issue.

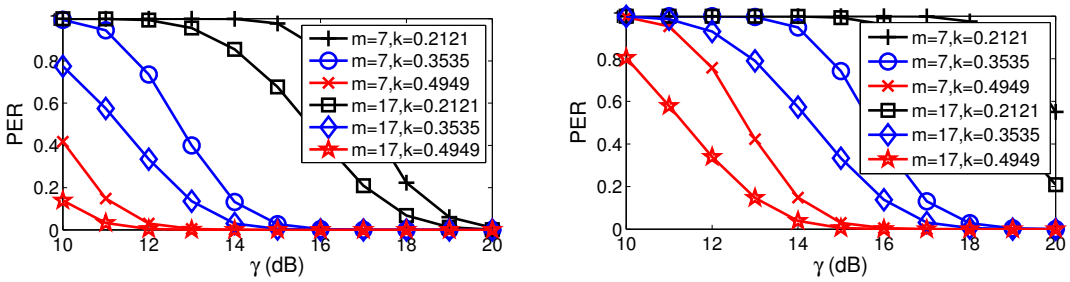
Automatic Gain Control. Automatic gain control (AGC) is widely adopted in receivers to enable dynamic adjustment of receiving gain. For QPSK and QAM modulation, the specified level usually corresponds to the mean power of all the constellation points. In SpecGuard, since Scheme 1 and Scheme 2 modify 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 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.

2.7 Performance Evaluation

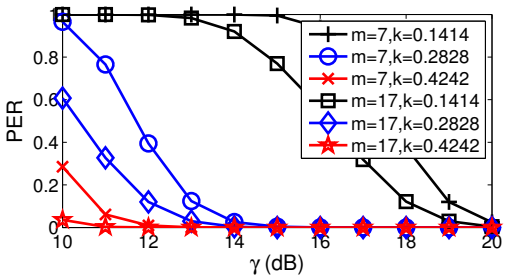
In this section, we evaluate SpecGuard using MATLAB simulations and USRP experiments. We also compare SpecGuard with [1] despite their different application scenarios.

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 of over 2,000 data packets, and each data point in USRP results represents an average across 10,000. It is worth noting that the numerical results based on our theoretical analysis



(a) Scheme 1.1

(b) Scheme 1.2



(c) Scheme 2

Figure 2.5: Permit Error Rates for Scheme 1 and Scheme 2.

in Section 2.5 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 [35], 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.

2.7.1 MATLAB Simulations

Fig. 2.4 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

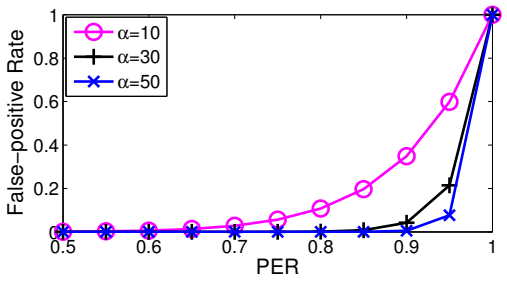


Figure 2.6: False-Positive Rate.

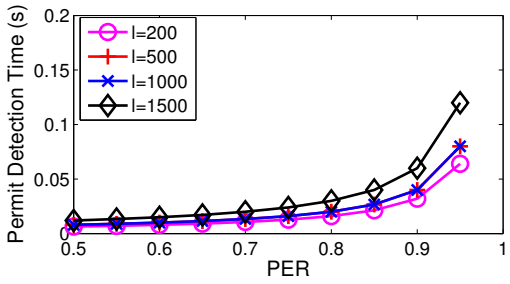


Figure 2.7: Average Permit Detection Time.

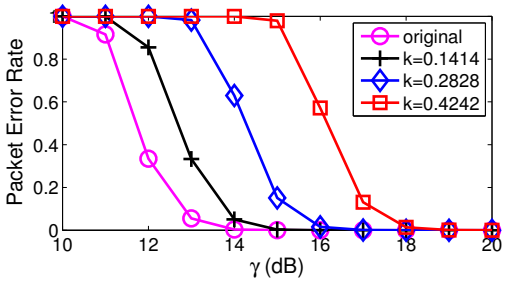


Figure 2.8: Data Error Rate for Scheme 2.

hard decision in all cases, so we focus on reporting the evaluation results based on the soft decision only due to space limitations.

Fig. 2.5 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. 2.5a and Fig. 2.5b show that Scheme 1.2 incurs a slightly higher PER than Scheme 1.1, which is consistent with the analysis in Eq. 2.1 and Eq. 2.2. 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.

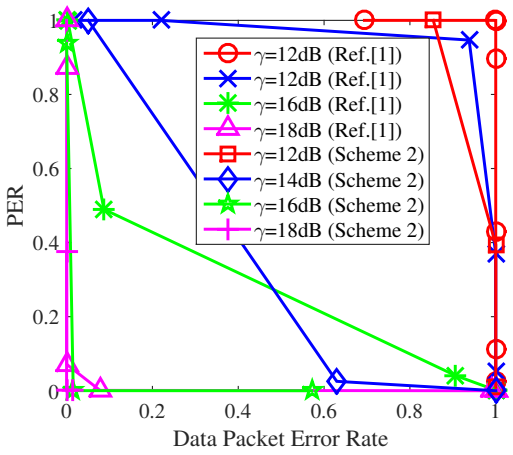
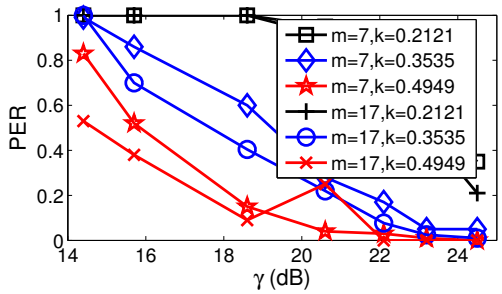


Figure 2.9: Comparison Between Scheme 2 and [1].

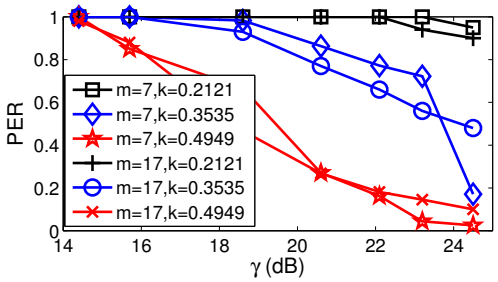
We also evaluated the PER for Scheme 3 in MATLAB. When γ equals $11-18$ dB, the PER is 1.00|0.99|0.92|0.66|0.31|0.07|0.02|0.00. This result highlights the superior permit detection performance of Scheme 3 in contrast to Schemes 1 and 2. One may note that all our schemes have very high PERs when $\gamma \in [10, 15]$ dB. As said above, $\gamma \in [10, 15]$ corresponds to very poor wireless channels over which normal data transmission are unlikely to occur [35]. In other words, all our schemes have sufficiently low PERs and work well in normal channel conditions.

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 2.5.1), where α is the number of verification attempts. Fig. 2.6 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 $\alpha = 5$.

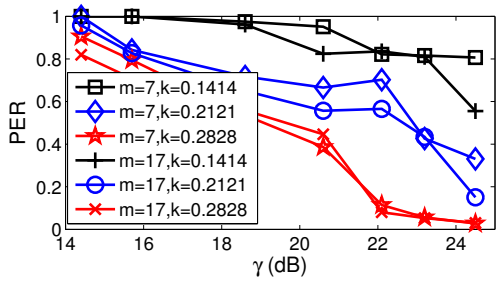
Moreover, we associate the results in Fig. 2.5 with the analysis in Section 2.5.2 to evaluate the fast property of SpecGuard. Here we let the data-transmission rate $r = 2$



(a) Scheme 1.1



(b) Scheme 1.2



(c) Scheme 2

Figure 2.10: PER Performance Using USRP.

Mb/s and the repetition parameter $m = 17$. Fig. 2.7 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 size data packet 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

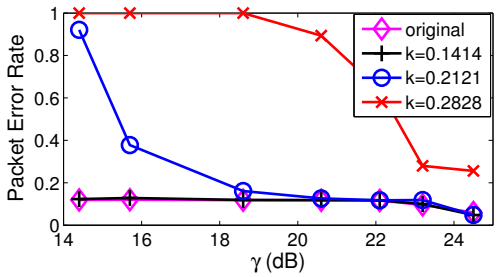


Figure 2.11: Data-Packet Error Rate for Scheme 2 Using USRP.

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. 2.8. Generally, the larger k, the more data-packet errors due to the reduced minimum distance between data-constellation points (cf. Fig. 2.2). Scheme 2 still works well for high SNRs.

Table. 2.1 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 lower 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 [1] in Fig. 2.9. In the comparison, we fixed $m = 7$ and varied the value of

Table 2.1: 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%

k. For [1], 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 [1].

2.7.2 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. 2.10 shows the PER for Schemes 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. 2.11, 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.

2.8 Conclusion

In this chapter, 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.

Chapter 3

SAFEDSA: A TEMPORAL APPROACH FOR SPECTRUM MISUSE DETECTION

3.1 Overview

In this chapter, we propose *SafeDSA* [7], a novel PHY-based scheme for authenticating secondary users in DSA systems. The key novelty of SafeDSA is to embed the spectrum permit into the cyclic prefix of each physical-layer symbol, which refers to prefixing a symbol with a repetition of the end. The cyclic prefix is widely used in wireless communication systems to eliminate the inter-symbol interference from previous symbols and simplify frequency-domain processing in multipath environments [32]. In SafeDSA, the secondary user increases (or decreases) the cyclic-prefix length in each symbol of a physical-layer frame if the next permit bit is 0 (or 1). A complete spectrum permit is transmitted via consecutive frames and can be easily decoded and then authenticated by a verifier interpreting dynamic cyclic-prefix lengths.

SafeDSA is theoretically analyzed and evaluated through detailed MATLAB simulations and USRP experiments. We show that SafeDSA has the following salient features that make it ideal for authenticating secondary users in DSA systems.

- **Robust**: SafeDSA can detect spectrum misuse with a maximum false-positive rate of 0.091 and a negligible false-negative rate in USRP experiments.

- **Efficient**: The most intensive computation in SafeDSA is estimating the cyclic-prefix length, which can be done very efficiently and usually two orders of magnitude faster than prior work [10]. SafeDSA is thus very feasible for both de-

dicated, resourceful verifiers [6] and resource-constrained mobile crowd-verifiers [8]. In addition, the communication overhead incurred by SafeDSA is negligible.

- **Non-intrusive**: SafeDSA requires minimal modification at the secondary user's physical layer and has negligible impact on normal data throughput. We also show that SafeDSA has negligible impact on channel estimation and frequency/timing estimation which rely on cyclic prefix.

The rest of the chapter is organized as follows. Section 3.2 introduces the system and adversary models. Section 3.3 outlines the background of the cyclic prefix and OFDM underlying SafeDSA. Section 3.4 details the SafeDSA design. Section 3.5 analyzes the theoretical performance of SafeDSA. Section 3.6 evaluates SafeDSA through detailed MATLAB simulations. Section 3.7 reports the performance of SafeDSA through USRP experiments. Section 3.8 concludes this chapter.

3.2 System and Adversary Models

3.2.1 System Model

SafeDSA consists of the following system entities.

- *Operator*: The SafeDSA operator can be a licensed spectrum owner or a spectrum-service provider managing many licensed spectrums. It issues spectrum permits to secondary users and may charge them accordingly. The operator instructs verifiers to detect fake secondary users. FCC designated a few TV white space (i.e., unused broadcast television spectrum) database administrators which allow secondary users to query TV white space availability based on time and location. These database administrators can naturally act as a SafeDSA operator.

- *Secondary user*: A secondary user needs to obtain a spectrum permit from the SafeDSA operator for using a given spectrum band at the desired location and time. A typical communication session involves a secondary transmitter and a secondary receiver. The secondary transmitter is the one to be authenticated and needs to embed its spectrum permit into physical-layer signals. So secondary-user authentication is equivalent to secondary-transmitter authentication.

- *Verifier*: A SafeDSA verifier is not engaged in data communication with the secondary user. Instead, it passively eavesdrops on the secondary user's transmission and tries to detect and verify a spectrum permit. Since the SafeDSA operations are very lightweight, verifiers can be either resourceful entities dispatched by the operator as in [6] or resource-constrained mobile crowdsourcing workers (referred to as mobile crowd-verifiers) as in [8].

An authentication instance in SafeDSA can be initiated either according to a pre-determined random schedule or when legitimate secondary users in a particular area report abnormal interference. The operator instructs one or multiple verifiers in a particular area to authenticate secondary users using a specific channel, and the instruction contains necessary information tied to the correct spectrum permit. Then the verifier tries to passively decode a spectrum permit from the secondary user's physical-layer signals, verifies it, and finally reports the authentication result to the operator. If a fake secondary user is detected, the operator can take further actions to stop spectrum misuse such as triangulating the fake secondary user and involving law enforcement. Note that the operator needs to know the locations of verifiers and also reward mobile crowd-verifiers. How to protect the location privacy of and to provide

incentives to mobile crowdsourcing workers have both been intensively studied and are orthogonal to this work.

3.2.2 Adversary Model

The attacker is a fake secondary user trying to use a spectrum band. He does not have a valid spectrum permit, so he has to fake one, repeat the one overheard from legitimate secondary transmission, or simply transmit without a spectrum permit. We assume that the attacker knows the entire SafeDSA operations and has full control of his radio transceiver to arbitrarily manipulate his physical-layer signals. We also assume that the attacker is computationally bounded and cannot break the cryptographic primitives used to generate the spectrum permit. Finally, we assume that the attacker cannot compromise the verifier, and the only solution to compromised verifiers is to use multiple verifiers.

3.3 OFDM and Cyclic Prefix

Orthogonal frequency-division multiplexing (OFDM) is a modulation technique which encodes digital data on multiple carrier frequencies. In contrast to traditional single-carrier communication systems, OFDM utilizes a group of closely spaced orthogonal sub-carrier signals to carry parallel data streams. For each subcarrier, the data information are modulated using a conventional modulation scheme such as quadrature amplitude modulation (QAM) or phase-shift keying (PSK). OFDM has become an extremely popular modulation technique used in digital audio broadcasting (DAB), digital television standard such as DVB-H, wireless LAN standards IEEE 802.11 a/g/n/ac/ad, LTE, and many other applications [36].

A cyclic prefix refers to prefixing a symbol with a repetition of the end. The concept traditionally roots in orthogonal frequency-division multiplexing (OFDM) [37],

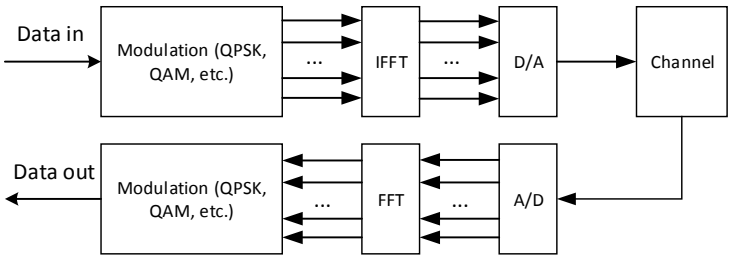


Figure 3.1: A Typical OFDM Framework.

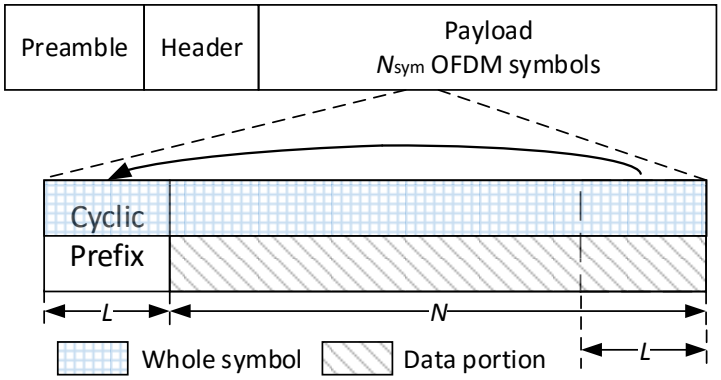


Figure 3.2: A General OFDM Frame Structure.

and now has its wide applications in single-carrier systems [38] as well to improve the resilience to multipath effect. Consider the typical OFDM framework in Fig. 3.1. After modulation of the input data bits, the individual samples are parallelized and then go through inverse fast Fourier transform (IFFT) to obtain samples in the frequency domain. The cyclic prefix is then added to form an OFDM symbol. Assume that N sub-carriers are used in OFDM, and let the symbol from the IFFT output be denoted by $\mathbf{x}' = [x[0], x[1], \ldots, x[N-1]]^{\mathsf{T}}$. Prefixing it with a cyclic prefix of length L, the resulting OFDM symbol is $\mathbf{x} = [x[N-L], \ldots, x[N-1], x[0], x[1], \ldots, x[N-1]]^{\mathsf{T}}$, where the last N samples compose the data portion. A general OFDM frame struc-

ture with N_{sym} OFDM data symbols is shown in Fig. 3.2 with the cyclic prefix added. The preamble is used for multiple functions such as signal detection, automatic gain control, frequency-offset estimation, and timing synchronization [39]. The frame header provides information about the frame length, coding rate, etc. Following the header symbols is the payload section, where N_{sym} OFDM data symbols are contained. Each OFDM data symbol can be further decomposed into $N+L$ samples, which is individually modulated using QPSK, QAM, or other techniques.

At the receiver, the cyclic prefix is removed before the data portion is processed, but it can serve a few important purposes. First, it eliminates the inter-symbol interference from the previous symbol as a guard interval. Second, it allows for simple frequency-domain processing, such as channel estimation and equalization, in multipath channels. Finally, it enables accurate timing and frequency synchronization at the receiver [40, 41].

The length of the cyclic prefix must be at least equal to the delay spread of the multipath channel, which can be interpreted as the difference between the time of arrival of the earliest significant multipath component and that of the latest multipath component. Legacy standards such as IEEE 802.11a/g specify a fixed long cyclic prefix (guard interval) of 800 nsec, which is equivalent to having $L = N/4$. In contrast, IEEE 802.11n can use a cyclic prefix of 400 nsec. IEEE 802.22 is the first cognitive radio-based international standard [42], in which the cyclic-prefix length can be set to 1/4, 1/8, 1/16 and 1/32 times the OFDM symbol length.

3.4 SafeDSA Design

In this section, we elaborate on the SafeDSA design, including how to construct, embed, extract and verify a spectrum permit.

3.4.1 Spectrum-Permit Construction

The spectrum permits in SafeDSA are similar to those in [6, 8, 10] and SpecGuard. A spectrum permit is issued by the SafeDSA operator to a secondary user for using a channel at a specified location and time. We assume that each channel of the SafeDSA operator has a unique *channel index*. In addition, we assume that the geographic region covered by the SafeDSA operator is divided into non-overlapping areas, each with a unique *area index*. Finally, we assume that all the wireless devices are loosely synchronized to a global clock.

A secondary user requests a spectrum permit by specifying a channel index, an area index, and a time duration which is assumed to compose $\kappa \geq 1$ equal-length time slots. The SafeDSA operator can use the efficient one-way hash chain technique to generate spectrum permits. Let $h(\cdot)$ denote a cryptographic one-way hash function such as SHA-1 [30]. The operator selects a random number n_κ and recursively computes $n_i = h(n_{i+1} \parallel \text{addr}_{\text{SU}}), \forall i \in [0, \kappa - 1]$, where addr_{SU} denotes the hardware address of the secondary user. Next, the operator sends n_κ securely to the secondary user through traditional security mechanisms such as TLS [31]. Finally, the secondary user recursively computes $\{n_1, \ldots, n_{\kappa-1}\}$ in the same way and uses $n_i, \forall i \in [1, \kappa]$, as his spectrum permit for slot i in the requested time duration.

Public-key methods can also be used to generate spectrum permits. In particular, the SafeDSA operator generates the spectrum permit for each slot $i \in [1, \kappa]$ of the requested time duration as its digital signature over $h(\text{addr}_{\text{SU}} \parallel \text{channel index} \parallel \text{area index} \parallel i)$ and send the κ spectrum permits securely to the secondary user. This method can enable proactive detection of fake secondary users at the cost of higher computational and communication overhead, which will be discussed in Section 3.4.3.

3.4.2 Spectrum-Permit Transmission

In this section, we illustrate how the spectrum permit is transmitted through and extracted from the cyclic prefix in SafeDSA. The cyclic-prefix length is usually designed as two to four times the root-mean-squared delay spread [43]. This level of redundancy ensures that the symbols will suffer the inter-symbol interference at a minimum possibility and also facilitates more accurate channel estimation and equalization. Under normal channel conditions, however, the cyclic-prefix length can usually be shortened to increase the throughput. We fully utilize this observation and embed the spectrum permit by dynamically changing the cyclic-prefix length according to the spectrum-permit bits.

It is worth noting that although the cyclic prefix exists in the preamble and header of an OFDM frame shown in Fig. 3.2, we start embedding the permit bits from the payload symbols. Maintaining the original cyclic-prefix length for preamble and header symbols makes timing synchronization and frequency-offset estimation easier and enables the secondary receiver to know the frame length before decoding the spectrum permit.

Although SafeDSA applies to any wireless technology using the cyclic prefix, we use the general OFDM frame structure in Fig. 3.2 for the ease of scheme description. We assume that the multipath channel has a delay spread of θ (measured in samples) and summarize the key design parameters in Table 3.1.

Permit Encoding

We embed the permit information by adaptively changing the cyclic-prefix length of the OFDM data symbols within a whole OFDM frame. In other words, one OFDM frame contains one permit bit, which enables more reliable detection of the permit

Table 3.1: Design Parameters.

N	Number of OFDM sub-carriers or size of FFT used
N_{sym}	Number of OFDM data symbols in an OFDM frame
L	The cyclic-prefix length (measured in samples)
θ	Channel delay spread (measured in samples)
m	Expansion ratio of the cyclic-prefix length
n	Compression ratio of the cyclic-prefix length

bit. Let $m(\geq 1)$ denote the expansion ratio of the cyclic-prefix length and $n(\leq 1)$ be the compression ratio of the cyclic-prefix length. The original cyclic-prefix length is L for each OFDM data symbol. When the permit bit to transmit is 0, the cyclic-prefix length expands to Lm; when the permit bit to send is 1, the cyclic-prefix length becomes Ln. Obviously, to protect the transmission from inter-symbol inteference, n needs to be strictly larger than θ/L. The mapping of the permit bits is illustrated in Fig. 3.3. Generally, we label the scheme as an M-ary scheme if the arity for permit-bit(s) embedding is M. Here, we embed only one permit bit ($M = 2$) inside each OFDM frame for ease of representation. We will demonstrate later that SafeDSA is easily extensible to higher arity such as $M = 4$. For the rest of the chapter, M is 2 unless otherwise stated.

We first analyze the parameter constraints. We define the probability of the permit bit being 0 or 1 as p_0 or p_1, respectively. Obviously, the data throughput would be increased or decreased if the cyclic-prefix length is reduced or extended, respectively. To avoid decreasing the data throughput, we require $p_0 m + p_1 n \leq 1$.

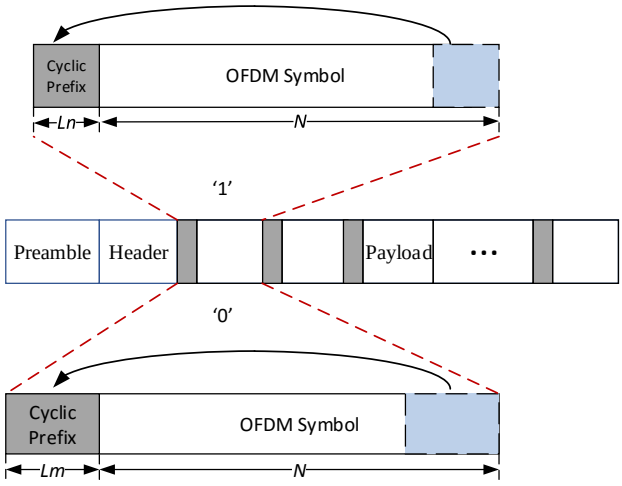


Figure 3.3: Mapping of Permit Bits to Cyclic-Prefix Lengths ($M = 2$).

Hence, by assuming that p_0 and p_1 are both 0.5, we have the following constraint set for parameter configurations:

$$\begin{aligned} & m + n \leq 2, \\ & m \geq 1, \frac{\theta}{L} < n \leq 1, \\ & mL, nL \in \mathcal{Z}. \end{aligned} \tag{3.1}$$

The second equation essentially adds limitations on how small n can be by requiring that the reduced cyclic-prefix length be no smaller than the delay spread of the multipath channel.

Permit Decoding

Permit decoding, or equivalently estimating the cyclic-prefix length from the received OFDM frame, relies on the dependency between the cyclic prefix and the matching end of the data portion. In the transmitted signal, the cyclic prefix is exactly the same as the matching end of the data portion. Although such ideal data dependency

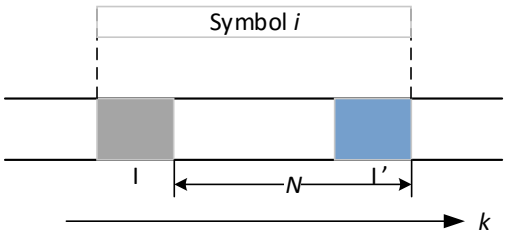


Figure 3.4: Data-Dependency Evaluation of OFDM Symbols.

is likely to be broken by inter-symbol interference and channel noise, the dependency is still expected to be very high. We use symbol i in the payload field of the OFDM frame to illustrate data-dependency evaluations, as shown in Fig. 3.4. Let $\mathcal{I}$ denote the set of the sample indices of the cyclic prefix and $\mathcal{I}'$ the set of indices of the data samples that are copied into the cyclic prefix. We denote the samples collected in serial by $\mathbf{r}$. The samples in the cyclic prefix and their copies are hence $r(k),\ k \in \mathcal{I} \cup \mathcal{I}'$. Our data-dependency evaluations are based on the pairwise correlation in the cyclic prefix [41]:

$$\forall k \in \mathcal{I}:\ \mathbb{E}\{r(k)r^*(k+p)\} = \begin{cases} \sigma_s^2 + \sigma_n^2 & p = 0 \\ \sigma_s^2 e^{-j2\pi\varepsilon} & p = N \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

In the above equation, σ_s^2 and σ_n^2 denote the average power of the signal and the noise, respectively; ε denotes the frequency difference in the transmitter and the receiver oscillators as a fraction of the intercarrier spacing. Note that the remaining samples $r(k),\ k \notin \mathcal{I} \cup \mathcal{I}'$ are mutually uncorrelated.

The above pairwise correlation is used in our model to facilitate the estimation of the cyclic-prefix length. However, one unique challenge is that the cyclic-prefix length varies according to the current permit bit. In other words, the sizes of the sets $\mathcal{I}$ and $\mathcal{I}'$ keep changing for each OFDM frame. Since the amplitudes of the

time-domain samples vary in a large range due to high peak-to-average power ratio (PAPR) frequently found in OFDM systems, simply adjusting sample amplitude by a uniform scale may not work. Therefore, to evaluate the likelihood of two cyclic-prefix lengths, we must ensure that the samples used are of the same lengths or normalized.

Based on Eq. (3.2), we consider three metrics for evaluating the data dependency within the estimated frame range. Let $L' \in \{mL, nL\}$ denote the candidate cyclic-prefix length. The first metric is the euclidian distance D, defined as:

$$D = \sum_{p=0}^{N_{\text{sym}}-1} \sum_{k=1}^{nL} |r((N+L')p+k) - r((N+L')p+k +N)|, \ L' \in \{mL, nL\}. \tag{3.3}$$

The second metric is the correlation C, defined as:

$$C = \sum_{p=0}^{N_{\text{sym}}-1} \sum_{k=1}^{nL} r((N+L')p+k) r^*((N+L')p+k +N), \ L' \in \{mL, nL\}. \tag{3.4}$$

According to Eq. (3.2), the smaller D or the larger C, the higher the likelihood that the candidate cyclic-prefix length L' is used in the received OFDM frame. Note that for both metrics, the number of samples used in one OFDM symbol is nL, which is the smallest possible cyclic-prefix length. In this way, the total number of pair-wise values added is the same, no matter which cyclic-prefix length is used in practice. It ensures that in one of the two possible cases of L', the samples always fall into the cyclic-prefix section.

One potential limitation shared by the above two metrics is that the number of samples used for evaluation per OFDM symbol is always nL, even though more samples are available (i.e., mL samples in the case of the permit bit being 0) to increase the estimation accuracy. To address this limitation, we further propose another evaluation metric T, which is inspired by the timing metric proposed in [2].

Specifically, we first define

$$P(p) = \sum_{k=1}^{L'} r((N+L')p+k)r^*((N+L')p+k \\ +N),\ L' \in \{mL, nL\},\ p \in [0, N_{\text{sym}}-1] \tag{3.5}$$

as the sum of L' correlations of sample pairs in one OFDM symbol. We also define the total sample energy within the corresponding cyclic-prefix section as

$$R(p) = \sum_{k=1}^{L'} |r((N+L')p+k)|^2, \\ L' \in \{mL, nL\},\ p \in [0, N_{\text{sym}}-1]. \tag{3.6}$$

The metric T is then defined as

$$T = \frac{|\sum_{p=0}^{N_{\text{sym}}-1} P(p)|^2}{(\sum_{p=0}^{N_{\text{sym}}-1} R(p))^2}, \tag{3.7}$$

which measures the correlation of the received samples after normalization. Since metric T uses different numbers of samples for different candidate cyclic-prefix lengths, a lower permit-detection error rate can be achieved. The detailed evaluations of all three metrics are postponed to Section 3.6.

The secondary receiver or the permit verifier can thereby apply either metric and obtain the estimated cyclic-prefix length by

$$\hat{L} = \text{argmax}_{L'}|C| \text{ or } \hat{L} = \text{argmin}_{L'}D \text{ or } \hat{L} = \text{argmax}_{L'}T\ , \tag{3.8}$$

which is mapped into a permit bit. After estimating the cyclic-prefix length, the secondary receiver removes the cyclic-prefix part and continues to decode the data symbols. In contrast, the verifier buffers all the estimated permit bits to construct and verify a candidate spectrum permit later.

A few issues are worth mentioning here. First, the index of $\mathbf{r}$ in Eqs. (3.3) to (3.6) starts from the first sample in the payload field, which implicitly assumes that

the timing offset correction can be achieved perfectly. This assumption may not hold in practice, and we discuss how to relax it in Section 3.7. Second, the detections of permit bits in different OFDM frames are independent from each other. Finally, an incorrect estimation of the cyclic-prefix length or permit bit results in a decoding error of data symbol inside the OFDM frame due to the removal of wrong cyclic-prefix sections. It is thus required that permit-bit detection be robust with a much lower error probability than that of the normal data transmission. We will demonstrate the effectiveness of this mechanism in Section 3.6 and Section 3.7 with MATLAB simulations and USRP experiments, respectively.

Extension of M

SafeDSA can be easily extended to higher arity encoding of the permit bits. As with the case of $M = 2$, we still need to apply the constraints as defined in Eq. (3.1) when M is larger, i.e., the channel conditions and the impact on normal data throughput still need to be considered. Here, we give a brief example of $M = 4$ in Fig. 3.5. The original cyclic-prefix length is L and the four candidate cyclic-prefix lengths are $L_1 \sim L_4$. The numbers in bracket are the Gray codes in which adjacent symbols differ by one bit. In this way, two permit bits can be embedded in one OFDM frame.

3.4.3 Spectrum-Permit Authentication

The SafeDSA operator activates spectrum-permit verification (or equivalently secondary user authentication) either according to some random schedule or when the legitimate user complains about severe interference. To do so, it chooses some verifiers in the specific area to ensure sufficient area coverage and sends them the channel index and the starting time of the legitimate secondary user's time duration through traditional TLS-like security mechanisms. If a one-way hash chain is used to con-

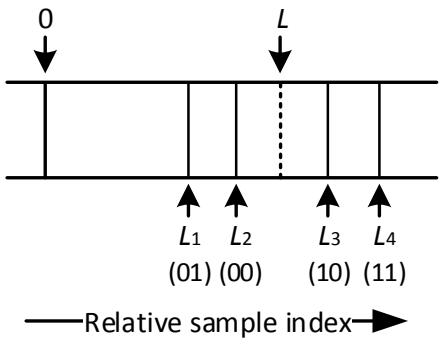


Figure 3.5: Extension of M With Gray Coding.

struct the spectrum permits, the operator additionally sends the hash value n_0 (see Section 3.4.1) to each chosen verifier. If the public-key method is chosen instead, nothing else needs to be sent.

After receiving the authentication request, a verifier first determines the current slot number based on the starting time of the specified time duration. Then it attempts to decode the permit bits on the specified channel as in Section 3.4.2. All the bits detected in the same time slot are concatenated in sequel. Since the verifier may have missed some permit bits of the current slot, it starts permit verification from the next slot. Consider slot i as an example. Assume that the decoded permit bits for slot i are $\{b_1, b_2, \ldots, b_w\}$ and that a spectrum permit is of β bits. The verifier executes the following verifications in order.

- Check whether $w \geq \beta$. This step is to make sure that at least one spectrum permit has been embedded.

- $\lfloor w/\beta \rfloor$ segments of spectrum permit can be detected in the time slot: $\langle b_{j\beta+1}, b_{j\beta+2}, \ldots, b_{j\beta+\beta} \rangle$ for $j \in [0, \lfloor w/\beta \rfloor - 1]$. Ideally, these segments should all be the same because

the same spectrum permit for slot i should be repeatedly sent. In practice, due to channel effects, they might vary. So as long as one segment of spectrum permit is correct, SafeDSA considers the secondary user legitimate. Note that if w is not an integer multiple of β, we simply abandon the bits $\langle b_{\lfloor w/\beta \rfloor \beta+1}, \ldots, b_w \rangle$. Let n'_i denote the candidate spectrum permit decoded sequentially. The verifier repeatedly performs the next step of operation until either the verification succeeds or all $\lfloor w/\beta \rfloor$ segments have been checked but fail the verification.

- If the one-way hash chain is used for spectrum permits, recursively computes $n'_j = h(n'_{j+1} \parallel \text{addr}_{\text{SU}}), \forall j \in [0, i-1]$ and verifies whether $n'_0 = n_0$. If the public-key method is used for spectrum permits instead, verify whether n'_i is the SafeDSA operator's digital signature over $h(\text{addr}_{\text{SU}} \parallel \text{channel index} \parallel \text{area index} \parallel i)$.

If the verification fails for all $\lfloor w/\beta \rfloor$ candidate spectrum permits, the verifier considers the secondary user fake. The authentication results are reported to the SafeDSA operator. If any fake secondary user is reported, the SafeDSA operator can dispatch some personnel to do some field test to physically locate the illegitimate secondary user and then stop spectrum misuse by possibly involving law enforcement.

To deal with possible synchronization errors between the secondary user and the verifier, the verifier can prefix $\{b_1, b_2, \ldots, b_w\}$ with the last Δ permit bits of slot $i-1$ and postfix them with the first Δ permit bits of slot $i+1$. The secondary user is authenticated for slot i as long as any consecutive β bits pass the above verifications.

To further mitigate permit-bit errors, we can encode the spectrum permit with an error-correcting code such as the Reed-Solomon code, in which case the second step above needs to contain error-correction operations before verifying the bit segments.

Also note that the public-key method for spectrum permits can enable the verifiers to proactively authenticate nearby secondary users without the operator's instructions. This can potentially lead to faster detection of fake secondary users at the cost of slightly higher computational overhead to verify a digital signature and higher communication overhead for transmitting longer spectrum permits to legitimate secondary users.

There is no way to prevent a legitimate user from sharing his spectrum permit with other users. Such cases are not considered spectrum misuse because only one spectrum user with a valid spectrum permit can use the channel at any time instant. Such spectrum-permit sharing can actually be helpful in a communication session involving multiple users who all need to embed a valid spectrum permit into their respective physical-layer signals. To accommodate this situation, we can let one secondary user purchase the spectrum permits and share them with other users through traditional TLS-like security mechanisms.

3.5 Analysis

In this section, we analyze the computational complexity of SafeDSA, its impact on channel estimation and frequency/timing offset estimation, and its security.

3.5.1 Computational Complexity

The computational complexity should be very low so that the authentication operations can be performed by both dedicated, resourceful verifiers and resource-constrained mobile crowd-verifiers, as the latter can be in large quantity to ensure more coverage and faster detection of fake secondary users. In SafeDSA, the most time-consuming operation is to estimate the cyclic-prefix length based on the data-dependency test. Since there are only two possible cyclic-prefix lengths for bits zero

and one, respectively, the computational overhead is trivial. In the closest work, FEAT [10], the verifier has to perform blind parameter estimation on multiple parameters of the OFDM signal, resulting in a high computation complexity. More specifically, to decode one permit bit, FEAT involves three major steps: symbol synchronization, frame synchronization, and frame frequency estimation. The symbol synchronization is the most computationally intensive part, in which all the possible samples in the cyclic-prefix sections are used to estimate the sample offset, IFFT size, and the cyclic-prefix length. For complete blind estimation, the possible ranges of the parameters to be estimated need to be comprehensive, covering all possible values. Let $|R_1|$, $|R_2|$, and $|R_3|$ denote the size of the estimation ranges for the three parameters. Then the complexity for symbol synchronization is $\mathcal{O}(|R_1||R_2||R_3|n_s)$, where n_s is the number of received OFDM samples. To give a concrete example, $|R_1|$, which stands for the range of the possible sample offset, needs to cover the whole range from 0 to $N + L - 1$. Likewise, the computational complexity of the rest two steps can be similarly derived. In contrast, SafeDSA performs the cyclic-prefix length estimation frame by frame, utilizing only the possible samples in the cyclic-prefix sections. It incurs a computational complexity of $\mathcal{O}(n_s)$, which is usually at least several hundred times less than FEAT.

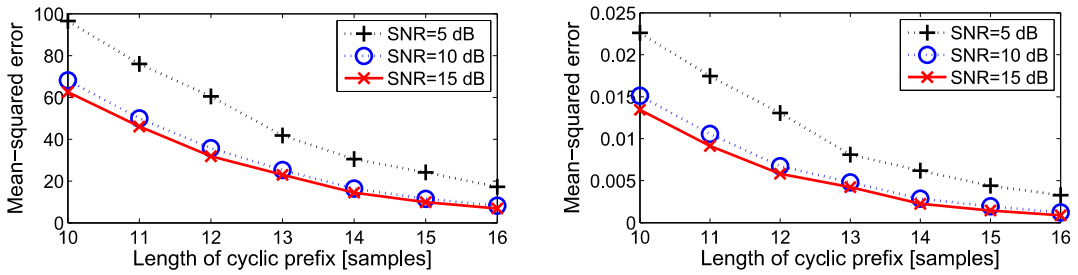
3.5.2 Impact on Channel Estimation

Channel estimation is indispensable to achieve coherent demodulation and consequently higher data rates. There has been numerous work dedicated to channel estimation for OFDM systems. In most work, training sequences or pilot sequences as included in the IEEE 802.11a standard are used for simple channel estimation [44, 45, 33]. Obviously, shortening the length of cyclic prefix will not have any impact on channel estimation if these mechanisms are adopted. There are some other work

such as [40] that utilizes discrete Fourier transform (DFT) for the channel estimation. Due to the repetition of the end of the symbol, it allows the linear convolution of a frequency-selective multipath channel to be modeled as a circular convolution, which in turn may be transformed to the frequency domain using the DFT. Since the estimation relies on the DFT property but not the cyclic-prefix length, shortening the cyclic-prefix length still does not have any negative impacts on channel estimation.

3.5.3 Impact on Frequency/Timing Offset Estimation

The unknown OFDM symbol arrival time and the mismatch of the oscillators in the transceiver pairs are the two major challenges in the design of OFDM receivers. To address these issues, previous work such as [46] relies on pilot symbols known to the receiver to perform the estimations. Similar to the channel estimation, the shortened cyclic-prefix length will not impact the estimation performance. Other work such as [41] exploit the cyclic prefix preceding the OFDM symbols, thus reducing the need for pilots. In 802.11a and other standards, the pilots are still used, hence making it less likely to purely rely on the cyclic prefix for frequency or timing offset estimation. Here, since the cyclic-prefix length is shortened, it is desirable that we can fully evaluate the impact of the change so that we are confident about its application in a majority of scenarios. Adopting the assumptions in [41] that no additional pilot carriers are inserted, we theoretically analyze the impact of shortening the cyclic-prefix length on the estimation of time and frequency offset. We choose the following parameter configurations for the evaluation: $N = 64$, $L \in [10, 16]$. The frequency offset denotes the difference in the transmitter and receiver oscillators as a fraction of the inter-carrier spacing ($1/N$ in normalized frequency) and set as 0.25. The channel simulated is an AWGN channel with different SNR values (5, 10, and 15 dB). The performance of the time and frequency estimators is shown in Fig. 3.6 in the form of



(a) Time Estimation.

(b) Frequency Estimation.

Figure 3.6: Performance of the Time and Frequency Estimators for the AWGN Channel.

mean-squared error. The evaluation metric is not normalized. It is clear that with higher SNRs, the estimators can achieve better performance. When the cyclic-prefix length is reduced from 16 to 14, the performance degradation is very limited. The degree of performance degradation increases with the cyclic-prefix length shorter than 12. Overall speaking, even when the cyclic-prefix length is 12, the estimators can still achieve a good performance with mean-squared error in time estimation being less than 40 and mean-squared error in frequency estimation being less than 0.08.

To summarize, when mechanisms such as [41] are adopted for time and frequency estimation, shortening the cyclic-prefix length in a controlled manner will lead to negligible negative impact. For other mechanisms estimating time and frequency offset without using the cyclic prefix, shortening the cyclic-prefix length does not have any influence on the estimation accuracy at all.

3.5.4 Security

For the security analysis, we focus on the attacks related to the spectrum permit. The attackers can either try to emulate the authorized transmitter or replay an overheard spectrum permit. SafeDSA is resilient to both attacks.

Emulation Attack

In the emulation attack, the attacker tries to spoof the verifiers nearby by generating a fake spectrum permit and embedding it into the cyclic prefix. The probability for a successful emulation attack is almost negligible due to the cryptographic primitives adopted. Recall in our design, an efficient hash chain or the public-key method is adopted to construct the spectrum permits. Therefore, without the root of the hash chain or the SafeDSA operator's private key, it is beyond the computational capability of the attacker to derive a spectrum permit based on observations of the authorized transmitters' signals or other rules that he might want to use.

Replay Attack

It is highly possible that the attacker can first decode the spectrum permit from the authorized transmitter's signal and then replay this spectrum permit for his own transmission. To deal with this attack, the key idea is to ensure that the spectrum permit is updated frequently. In SafeDSA, each time slot has a unique spectrum permit, so the intercepted spectrum permit will be invalid in subsequent time slots. The impact of replayed spectrum permits can thus be reduced by using smaller slot length at the cost of higher computational overhead at the verifier. In addition, it is still possible to identify the attacker even within the same time slot. For example, the verifiers can associate the signal characteristics (SNR, RSSI, directionality, etc.) with the secondary user. When any inconsistent feature appear, the verifiers could generate an alarm report for the operator to further investigate the issue.

3.6 MATLAB Simulations

In this section, we conduct thorough evaluations of SafeDSA in MATLAB. Since improving the throughput of data transmission is not the major goal of our scheme,

we simply let $n+m=2$, meaning that we maintain the average cyclic-prefix length as the original cyclic-prefix length by assuming that bits one and zero are equally likely to appear in the spectrum permit. We define a new metric for ease of representation, called the deviation of cyclic-prefix length:

$$d = L - nL = mL - L. \tag{3.9}$$

This new metric essentially quantizes the amount of variation of the cyclic-prefix length for the payload symbols. We aim to investigate the impact of d on permit detection.

Below, we first fully study the impact of each parameter in SafeDSA and then compare SafeDSA with FEAT [10] and SpecGuard [8]. We do not choose Gelato [6] for comparison because it reduces the normal data throughput in contrast to SafeDSA, FEAT, and SpecGuard. The default simulation parameters are as follows: $N = 64$, $L = 16$, and $N_{\text{sym}} = 25$. For most simulations, the AWGN channel is used unless otherwise stated, and the modulation scheme for each OFDM sub-carrier is QPSK. The cryptographic function used for the construction of the spectrum permit is the SHA-1 function, which generates a 160-bit value.

Since the secondary receiver needs to correctly estimate the cyclic-prefix length for decoding the data portion, it is extremely important that permit-bit detection is robust and reliable. In extreme cases where most data packets would fail such as low SNR cases, we also want to ascertain that it is not because permit-bit detection fails.

3.6.1 Data-Dependency Metric

Recall that in Eq. (3.3), Eq. (3.4) and Eq. (3.7), three evaluation metrics have been proposed. C relies on the correlation; D calculates the Euclidian distance; and T performs the normalization based on received sample energy for the correlation. We

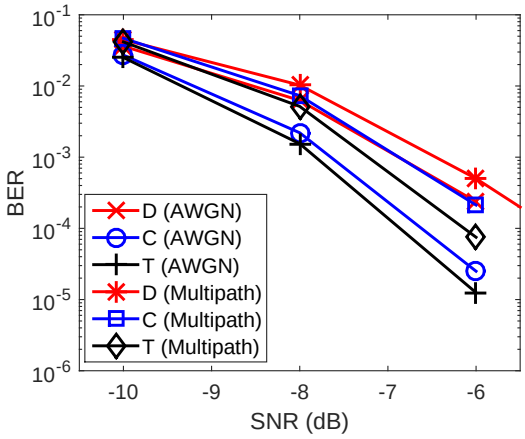


Figure 3.7: Comparing Data-Dependency Metrics (d=1).

compare the performance of the three metrics with different channel types: AWGN and multipath Rayleigh fading channel with five channel taps. The results are shown in Fig. 3.7. Clearly T outperforms the other two metrics in both channel conditions. This is expected because the metrics C and D adopt fewer samples and also because the variations of sample amplitudes are not averaged. Generally, in both channel conditions, SafeDSA can achieve very good permit-detection performance even in very low SNR ranges. For the rest of the simulations and experiments, unless otherwise noted, T is chosen as the data-dependency-test metric.

3.6.2 Deviation of the Cyclic-Prefix Length

We evaluate the impact of d, the deviation of the cyclic-prefix length. The larger d is, the higher requirement the system has over the channel because the shortened cyclic-prefix length becomes less resilient to inter-symbol inteference. A natural question would be that whether increasing d can lead to better permit-detection performance. We conduct evaluations by changing the value of d for all the three evaluation metrics and observe similar phenomena. Fig. 3.8 illustrates the result using T. The

result is somehow counter-intuitive. The permit BER is the lowest when $d = 1$, while d being 2 leads to the worst simulation result.

This motivates us to think deeper and find out the real cause behind this phoenomenon. Recall that the intuition behind designing the data-dependency metrics in Eq. (3.3), Eq. (3.4), and Eq. (3.7) is that only the matching samples in the cyclic-prefix section can be largely dependent and thus achieve a small or large value as specified in Eq. (3.8). In other words, the more samples wrongly picked outside of the cyclic-prefix sections for the wrong candidate cyclic-prefix length, the better we can distinguish the two candidate cyclic-prefix lengths. Therefore, we performed another set of data analysis to prove the correctness of the above conjecture and matches the analysis to this seemingly wrong result in Fig. 3.8. In this analysis, we consider one case where the true cyclic-prefix length is larger than the estimated one. In this case, we count how many samples are considered potential cyclic-prefix samples but in fact data samples. The result is as follows. When d is 1, the number is 311; when d equals 2, the number is 256; and when d is 3, the number is 263. Since the case where d is 1 has the largest number of misaligned samples, it is surely easier to be distinguished than other cases. Besides, this analysis matches the result in Fig. 3.8, which proves its correctness.

Based on the above analysis, we can draw the conclusion that permit-detection performance is largely dependent on how many misaligned samples are used for the data-dependency test rather than d itself. However, we do not think that it is necessary to propose a guideline to demonstrate how we can directly manipulate the main factor mentioned, as Fig. 3.8 shows that the three curves are close to each other with a low BER overall. For the rest of the chapter, unless otherwise mentioned, d is set to 1.

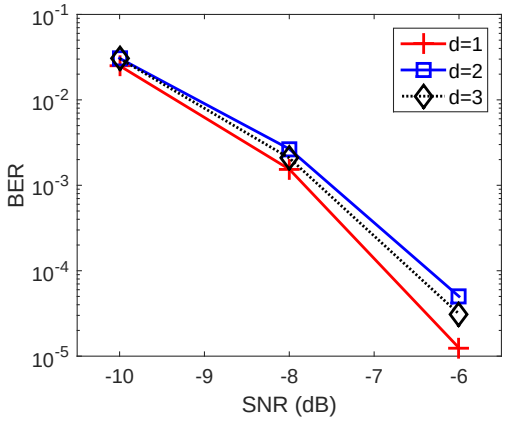


Figure 3.8: Permit BER for Different ds.

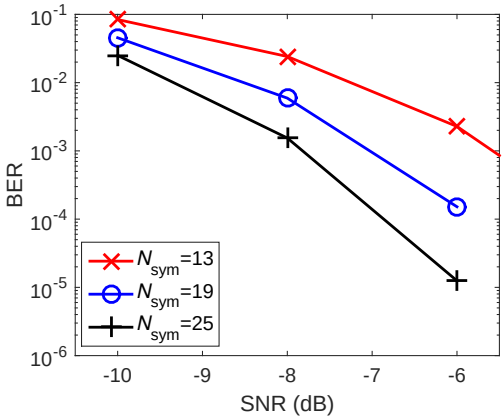


Figure 3.9: Permit BER for Different Frame Lengths.

3.6.3 Frame Length

SafeDSA relies on a whole frame of the received samples to perform the cyclic-prefix-length estimation. Therefore, we are curious about how many samples are good enough for the cyclic-prefix-length estimation. In this test, we change the value of N_{sym}. According to previous results, we know that permit-detection performance has been very reliable when N_{sym} is 25. Also, intuitively speaking, the larger N_{sym}, the more samples that can be collected for the cyclic-prefix-length estimation, the more accurate the cyclic-prefix-length estimation, and hence the lower the permit BER. Fig. 3.9 shows the simulation result when N_{sym} varies from 13 to 25. We clearly see the trend that larger N_{sym} can deliver better permit-detection performance. In addition, even when N_{sym} is 13, which corresponds to about 156 bytes per frame (packet), the BER can be as low as 3×10^{-3} when SNR is -6 dB. Again, the effectiveness of SafeDSA has been proved.

Additionally, we show the false-positive rate of SafeDSA in Fig. 3.10. For ease of evaluation, we simply let $\lfloor w/\beta \rfloor$ in Section 3.4.3 be 1, i.e., only one copy of candidate

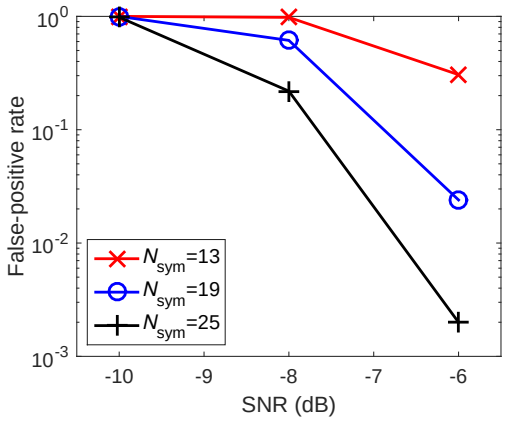


Figure 3.10: False-Positive Rate for Different Frame Lengths.

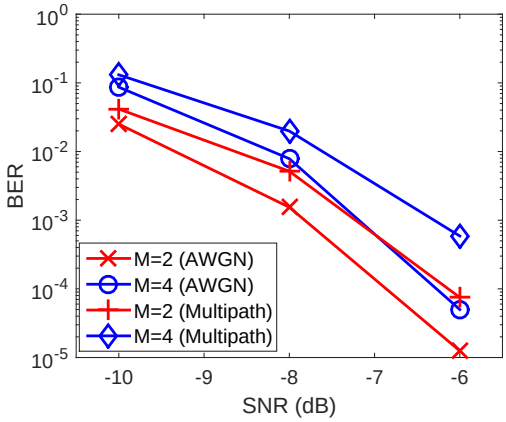


Figure 3.11: Permit BER for Different Ms.

spectrum permit is verified. A false positive (negative) refers to a legitimate (an illegitimate) secondary user mistaken for an illegitimate (a legitimate) user. We observed that the false-positive rate descends faster than the BER curves shown in Fig. 3.9. When SNR is -6 dB, the false-positive rate can be as low as 2×10^{-3}. As expected, the frame length plays a key role in the performance. When N_{sym} is 13, the false-positive rate is generally much worse than that when N_{sym} is 25. However, since usually wireless communications are conducted when SNR is above 0 dB [35], SafeDSA can achieve desirable detection performances in this regard. On the other hand, false-negatives occur in cases such as when the fake secondary user randomly guesses a correct spectrum permit. Since the spectrum permit is usually of hundreds of bits, the false-negatives can rarely happen.

3.6.4 The Value of M

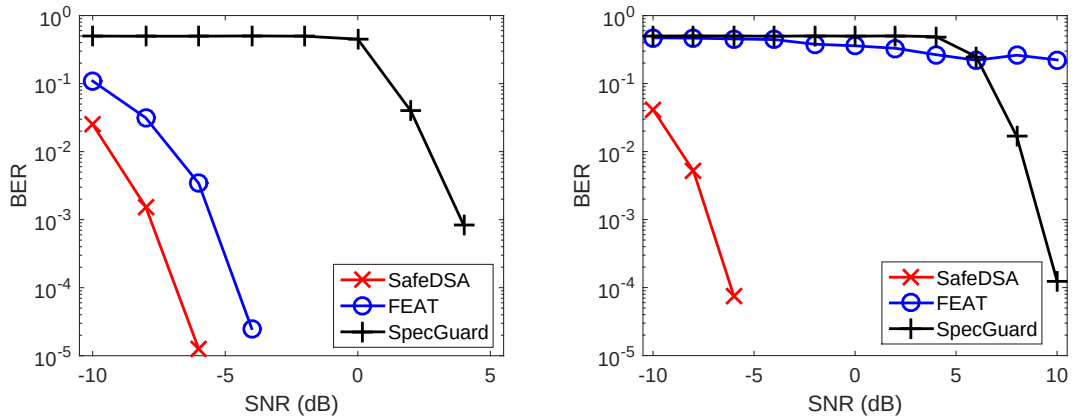
We also evaluate permit-decoding performance for different values of M in Fig. 3.11. For $M = 4$, we simply let $L_1 \sim L_4$ be 14, 15, 17 and 18. As expected, the permit

BER increases when M increases from 2 to 4. Still, the performance is quite good given that the SNR is so low.

3.6.5 Comparison With Related Work

In this section, we compare SafeDSA with FEAT [10] and SpecGuard [8]. In FEAT, the sampling frequency is set as 1 MHz. The maximum positive frequency offset that can be used to embed the authentication signal into a frame of the message signal f_a is 5 KHz. There are three schemes in SpecGuard [8], among which Scheme 1 increases the overall power consumption, and Scheme 3 requires additional trust relationship between the secondary transmitter and receiver. So we only use Scheme 2 in SpecGuard for comparion, with the amplitude boost factor k set as 0.14. According to the authors in [8], the additional power is 2% when k is 0.14, which is an acceptable overhead. We embed one permit bit for the entire OFDM frame for all the schemes. In addition, M is set as 2. Using the above configurations, we aim to conduct a fair comparison of these three schemes without assuming additional resources.

First, we consider two different channel types: AWGN and multipath Rayleigh fading channels. Fig. 3.12 shows the evaluation results. Clearly, SafeDSA outperforms FEAT and SpecGuard with a very robust permit-detection performance even under extremely low SNR contexts. In contrast, SpecGuard can generally provide a good performance when SNR is high enough, i.e., above 0 dB. This is usually good enough since in such low SNR cases, the data communication efficiency can be greatly influenced as well. FEAT also can perform reliably in the AWGN channel but fails to perform consistently well in the multipath Rayleigh fading channel. Even when SNR is 10 dB, the permit BER is around 22%. This is undesired, as it indicates a spectrum permit with usually a few hundreds of bits will be decoded wrongly at 100%. The



(a) AWGN Channel.

(b) Multipath Rayleigh Fading Channel.

Figure 3.12: Permit BER Comparison for Different Channels.

root cause of this failure, as we later discover, is that the estimation of some system parameters such as N_{sym} is wrong. This will cause incorrect alignment of samples and hence wrong frequency offset estimation. Note that we used 400 frames for the simulation for each iteration, which is usually long enough for one whole spectrum permit transmission. Certainly, if more frames are transmitted, the permit BER of FEAT could be improved, but FEAT still does not work well in multipath environments. By comparison, SafeDSA performs cyclic-prefix-length estimation individually for each frame, requiring minimum samples for the estimation. Therefore, SafeDSA has no requirement on the overall number of frames transmitted, and so is SpecGuard.

We then compare the three schemes for different frame lengths in the AWGN channel. Fig. 3.13 shows the results. As expected, the permit BERs all decrease with N increases, and both SafeDSA and FEAT can provide very reliable permit detection. Although SpecGuard fails to work when SNR is below 0 dB, the BER curve rapidly descends when SNR is over 0 dB. In short, the three schemes can all work well even when the frame length is small.

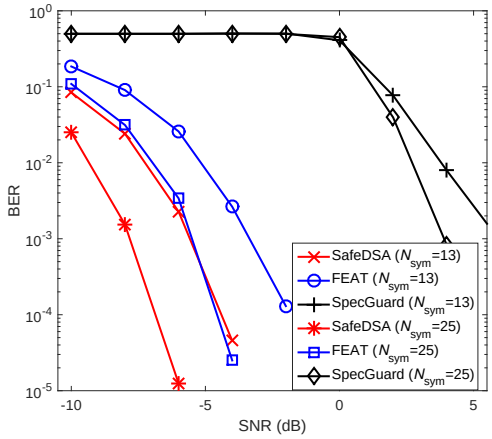


Figure 3.13: Permit BER Comparison for Different Frame Lengths.

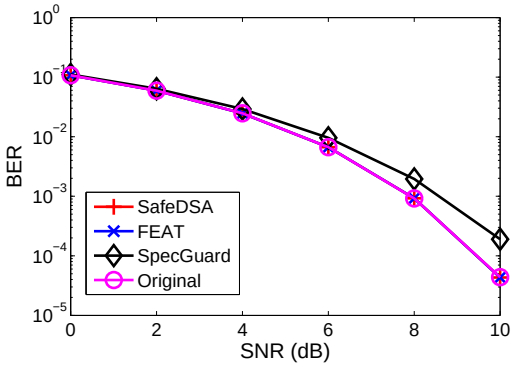


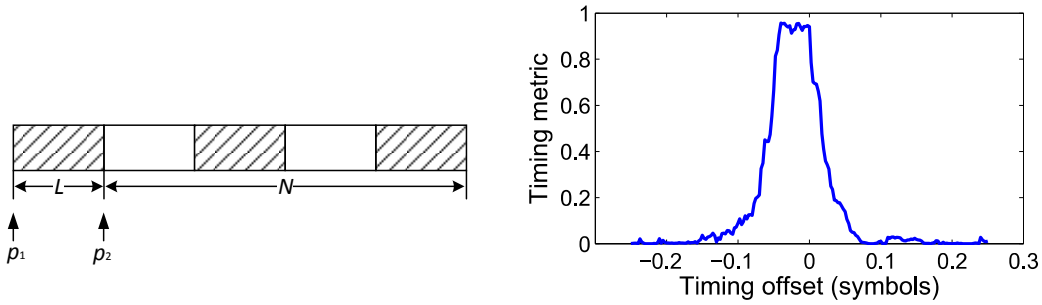
Figure 3.14: Data BER Comparison in AWGN Channel.

Lastly, we compare the three schemes' impact on normal data transmission. FEAT embeds the spectrum permit in the form of intentional frequency offset. As long as the overall frequency offset of the signal received is within a certain range that can be corrected by the secondary receiver, there is no negative impact on normal data transmission. SafeDSA essentially uses the timing gap between the "useful" payload information to embed the spectrum permit. Hence, as long as the timing gap, realized by the cyclic prefix in OFDM symbols, is longer than the delay spread of the channel, it also does not affect normal data transmission. In contrast, SpecGuard needs to decrease or increase the transmission power and thus may degrade the BER performance of normal data transmission in the latter case. The comparison is shown in Fig. 3.14, where the curves of FEAT and SafeDSA are strictly aligned with the original OFDM system's curve. The SNR ranges are selected as 0 to 10 dB with consideration of the higher BER of data bits compared with that of permit bits.

3.7 USRP Experiments

To fully understand how SafeDSA performs in practice, we further implement it in GNU Radio with USRP N210 as the hardware platform. In our experiments, we use three USRPs to represent the secondary transmitter, the secondary receiver and the verifier. The USRPs are separated from each other by around 3 meters. 48 out of the 64 OFDM sub-carriers are used for data transmission, and 4 sub-carriers are used for pilot-symbol transmission. The bandwidth of the signal is chosen as 1 MHz due to the hardware limitation. We adopt two preambles for timing and frequency offset-estimation. One additional symbol is assigned for the frame (packet) header information. The configuration of other parameters are the same as the default configurations in Section 3.6.

Different from the 802.11 standard introduced earlier, our USRP N210 transceiver only uses two preambles as discussed in [2] for the frequency and timing synchronization. The length of these two preambles is the same as the normal data symbols. The first symbol has identical halves in time domain, so the correlation between these two halves can be performed to find the timing metric as defined in the chapter at the receiver end. As discussed in Section 3.4, the cyclic-prefix length for the preambles as well as the packet header is the original one, i.e., one fourth of the FFT size in our setting. The permit-bit embedding starts from the first payload symbol and lasts until the last payload symbol inside the frame. Timing synchronization is achieved by using the special preambles defined in [2]. Hence, adopting the variable cyclic-prefix length for the payload does not affect frame synchronization. The decoded header provides the frame (packet) length information. The secondary receiver or the verifier then performs the cyclic-prefix-length estimation based on the frame (packet) length and accordingly removes the cyclic-prefix section of each symbol.



(a) The Type 1 Synchronization Symbol Used.

(b) The Plateau Effect (SNR = 20 dB).

Figure 3.15: The Plateau Effect When Using the Timing Offset Estimation Method in [2].

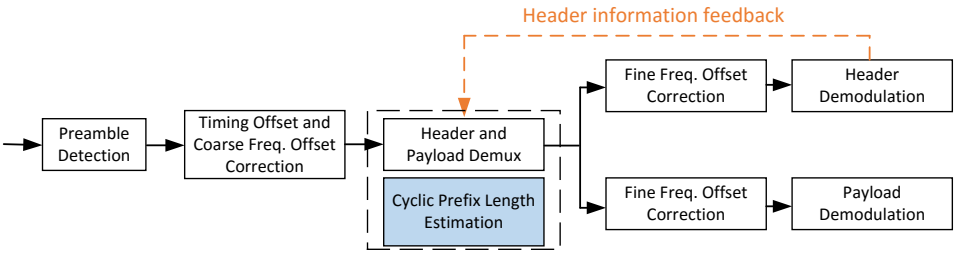


Figure 3.16: The Flowchart of the SafeDSA Receiver Design in GNU Radio.

Different from MATLAB simulations, which assume perfect timing and frequency synchronization, in our GNU Radio implementation, the synchronization is achieved by detecting the plateau as defined by the timing metric in [2]. Fig. 3.15a shows the Type 1 synchronization symbol used for the timing-offset estimation. The first grayed section with length L belongs to the cyclic-prefix section of the symbol. By designing the synchronization symbol as having two identical halves, essentially the three grayed portions are the same, and so are the two non-grayed portions. The timing offset estimation is conducted starting from a pointer p until involving N

samples. The plateau is reached when p is between p_1 and p_2, as shown in the figure. Fig. 3.15b shows the plateau effect when SNR is 20 dB. In the AWGN channel, the plateau has a width of the cyclic-prefix length due to the special preamble defined. The start of the frame can be taken to be anywhere in this window without a loss in the received SNR. This ambiguity of the start of the frame, however, makes it difficult to obtain the right samples for the cyclic-prefix-length estimation in Eq. (3.7). To address this issue, our receiver and verifier can conservatively use fewer samples than L to perform the estimation so that the samples used fall into the cyclic-prefix sections. When SNR is very low, another challenge to obtain the correct cyclic-prefix samples is that p could be out of the range between p_1 and p_2. Therefore, we increase the region of p from a single point to a region which spans across 7 samples before and after the original point. This slightly increases the computational overhead, which is nonetheless still much lower than FEAT, and the region only needs to be expanded in low SNR cases.

Fig. 3.16 illustrates the flowchart of the SafeDSA receiver design. The cyclic-prefix-length estimation module is the core module of SafeDSA and is added in the module of "header and payload demux." Before performing the fine frequency offset correction and demodulation for the payload section, the receiver needs to first wait for the feedback of the header information to obtain the frame length and other frame parameters. After the header is correctly decoded and parsed, the new "header and payload demux" module can first retrieve the corresponding samples and then perform the cyclic-prefix-length estimation. The payload section is extracted once the cyclic-prefix-length estimation is finished for the current frame. The verifier essentially shares similar designs with the receiver except that no payload processing such as fine frequency-offset correction and demodulation for payload is necessary.

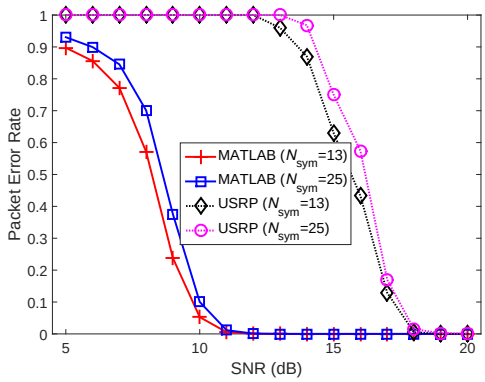


Figure 3.17: Packet Error Rate Comparison Using USRP Benchmark Transceivers and MATLAB Simulations.

To evaluate the performance of SafeDSA in real environments, we first show the packet error rate comparison using USRPs and MATLAB for the AWGN channel in Fig. 3.17. We vary N_{sym} from 13 to 25 to illustrate the impact of frame (packet) length. As expected, although generally the two curves of MATLAB simulations share the same trend with the two curves of USRP experiments, we observe an SNR offset of about 8 dB. The SNR offset can be caused by several factors: the inaccuracy of SNR estimation, the channel condition being more complicated in practice due to the multipath, fading, etc. Also, it could be that the benchmark OFDM transceivers using GNU Radio are relatively simple. The USRP equipments might also not be able to provide an optimal performance due to either the hardware limitation or the configurations of certain parameters. There might be other techniques that can be adopted to improve the performance such as using a longer preamble for timing, frequency offset estimation and channel estimation, a better filter to remove undesired noise and interferences, etc. The purpose of showing this comparison is to give readers a sense about how large the room is for improvement on our benchmark OFDM receiver and hence on our implementation for SafeDSA.

The permit BER based on USRP experiments are not shown here because for most SNR cases, the value is simply 0 or too low to observe. In our experiment, the SNR range (13 dB$\sim$20 dB) match with the range in Fig. 3.17 where the packet error rate is less than 1, which indicates that some data packets can be correctly decoded. We consider it not necessary to further degrade the SNR since in those extremely low SNR cases, the normal data transmission simply cannot be performed due to 100% packet error rate. We test four cases in total to evaluate the permit BER by varying N_{sym} from 13 to 25 and varying M from 2 to 4. As expected, when N_{sym} is 25, the permit BER is always 0 or too low to observe. When N_{sym} is 13, the permit bit errors are detected when SNR is below 15 dB. Specifically, the permit BERs are around 2×10^{-4} or 5×10^{-4} for $M = 2$ and 2×10^{-4} or 6×10^{-4} for $M = 4$ when SNR is 14 or 13 dB, respectively. The corresponding false-positive rate is 0.091 at maximum. This proves that permit-bit detection can be very reliable in practice. We, however, do notice that the MATLAB simulation results in Fig. 3.9 and Fig. 3.11 are still much better than the results listed here even when considering the SNR offset mentioned earlier. After a deeper investigation, we find that the root cause of this performance degradation is that in low SNR cases, the timing offset estimation our implementation adopts can have a large variation, which indicates misaligned samples are used for all the candidate cases. To alleviate this, possible solutions can be as follows: adopting a larger range of samples for the data-dependency test so that the range covers the real sample offset; and implementing a more robust timing offset estimation mechanism such as [47] to ensure consistently small sample offsets. These investigations are left as future work.

3.8 Conclusion

This chapter proposes SafeDSA, a novel PHY-based scheme using dynamic cyclic-prefix lengths to safeguard DSA systems against fake secondary users. In contrast to previous work, SafeDSA incurs no additional power consumption, is computationally efficient, and can detect fake secondary users with extremely low false-positive and false-negative rates in different channel conditions. The efficacy and efficiency of SafeDSA are confirmed by detailed MATLAB simulations and USRP experiments.

Chapter 4

PRIVACY-PRESERVING CROWDSOURCED SPECTRUM SENSING

4.1 Introduction

Database-driven DSA [22, 48] is the FCC-approved de facto paradigm. In such a system, a spectrum service provider (SSP) accepts registrations from PUs and determines spectrum availability, and SUs are all required to inquire the SSP about the availability of any interested spectrum before using it. Current SSPs estimate spectrum availability based on PUs' registered locations and transmission schedules in combination with radio propagation modeling. Some measurement studies such as [49, 50], however, show that such estimations are often inaccurate and tend to be overly conservative due to ignoring local environmental factors, resulting in a considerable waste of valuable spectrum resources. In addition, current spectrum databases cannot provide the quality information of channels, which can significantly vary in space and time. Moreover, the locations of primary and secondary users cannot be validated, so a spectrum database administrator may return wrong spectrum occupancy information to secondary users.

Crowdsourced spectrum sensing (CSS) is very promising for mitigating the drawbacks of the current spectrum databases. In this approach, a spectrum database administrator recruits distributed mobile users to sense a given channel around a specified location and decides the channel occupancy by aggregating the sensing results. The feasibility of CSS is backed up by a few trends. First, 497 million mobile devices were added in 2014, and global mobile devices will grow to 10 billion by 2019 at a CAGR of 8% [27], which implies sufficient geographic coverage especially in popu-

lated metropolitan areas where DSA systems are expected to play significant roles. Second, future mobile devices are very likely to be capable of spectrum sensing given the expected pervasiveness of DSA-based wireless systems [51]. Last, mobile devices are increasingly powerful in self-localization, communication, and computation, which has fostered the explosive popularity of mobile crowdsourcing applications [52]. With CSS in place, the spectrum database administrator does not need to deploy a dedicated large-scale sensor network for spectrum sensing.

A typical CSS system works as follows. The spectrum database administrator publishes spectrum-sensing tasks either periodically or randomly. Each spectrum-sensing task involves one or multiple channels, a pre-determined set of geographic locations, and the sensing time. The sensing results from the designated locations can be aggregated to jointly determine the channel occupancy at the specified time. Each mobile user in the CSS system can independently decide his capability of performing the sensing tasks. Given the participating requests, the spectrum database administrator can select a set of users for each sensing task.

There are many challenges for pushing the promising CSS system above into practice. For example, strong incentives must be provided to stimulate self-interested mobile users for spectrum sensing. Designing incentive mechanisms for CSS systems is a non-trivial task. On the one hand, different users may want different rewards for the same sensing task. For instance, a user far away from the allocated location may require more to compensate for his longer driving time and higher fuel consumption; a user may also lie about his travel distance to a specific sensing location to gain more. On the other hand, the spectrum database administrator wants to minimize the overall participants' cost (i.e., social cost) for any sensing task as long as the sensing quality is sufficient. Another significant challenge lies in the location privacy of mobile users. Since spectrum-sensing tasks involve rich spatiotemporal information,

the whereabouts of participating users can be easily exposed, thus discouraging mobile users wary of their location privacy.

4.2 Related Work

4.2.1 Location Privacy

There is a rich literature on location privacy in general frameworks, for which a nice review for location privacy-preserving mechanisms (LPPMs) can be found in [53]. In addition, a formal framework for the analysis of LPPMs is proposed in [54].

4.2.2 Privacy and Security in DSA Systems

There are some elegant schemes on location privacy in CSS systems [21, 22, 55, 56, 57]. The majority of the schemes focus on preventing the spectrum database administrator from inferring the physical sensing locations based on submitted sensing reports.

Some schemes aim to provide location-proof verification or privacy protection for centralized dynamic spectrum access [23, 58, 59, 60]. Our solutions are based on a crowdsourcing model, which is different from the aforementioned works.

Some other schemes aim to detect false sensing reports [18, 20, 61, 62, 63, 64] or spectrum misuse [7, 8, 10, 25]. Our work focus on the pre-sensing phase and is orthogonal to these nice efforts.

4.2.3 Location Privacy in Spatial Crowdsensing

Another line of work aims to address location privacy leakage in general spatial crowdsourcing systems [56, 65]. To *et al.* [56] proposed a framework to protect location privacy of workers during the task assignment phase. Different from [56], DPSense

does not need the trusted service provider as in their work to perform the sanitized database release and geocast of spatial crowdsourcing tasks. In addition, DPSense targets a totally different application scenario in which spectrum-sensing tasks have strict sensing time requirements. Pournajaf *et al.* [65] considerd spatial task assignment for crowd sensing with cloaked locations. Different from this work, DPSense considers completely different system models and involves the time constraint of the sensing tasks. The task scheduling and the probabilistic model for participants to accept/decline sensing tasks makes it challenging or impossible to adapt the scheme [65] to our application scenario.

4.2.4 Task Assignment

In addition, there is a surge of interest on task assignment in spatial crowdsourcing [66, 67, 68]. He *et al.* [66] seek to maximize the rewards of the platform with consideration of geographic locations and time budgets of mobile users. But they did not consider maximizing the task fulfillment ratio, which is a critical design objective in the context of CSS. Cheng *et al.* [67] aim to maximize both the spatial and temporal diversity of spatial crowdsourcing tasks but do not consider the minimization of travel distances. Deng *et al.* are the first to study the combination of task assignment and scheduling in spatial crowdsourcing [68], but their work differs from DPSense in two main aspects. First, the task assignment in [68] is based on known participants' locations and does not provide any location privacy guarantee. Second, the tasks in their model have deadlines such that participants can perform the tasks any time before the deadline. This is different in our scenario where spectrum-sensing tasks have strict requirement on the sensing time. Hence, it is non-trivial to directly extend these existing efforts to the context of CSS.

4.2.5 Incentive Mechanisms Design and Differential Privacy

Numerous efforts [69, 70, 71, 72] have been made on incentive mechanism design for crowdsourcing worker selection. Our work differs from this line of works by specifically addressing spectrum sensing and also location privacy.

Differential privacy [73, 74, 75] has emerged as a powerful tool to provide statistical guarantee of the data privacy with the trade-off of the data utility. Xiao *et al.* in [76] found that the well known l_1 norm sensitivity fails to capture the geometric sensitivity in the two-dimensional space and proposed a planar isotropic mechanism for the location perturbation, which is the first to achieve the lower bound of differential privacy in the specific application scenario. The work in [77, 78] targets differentially private spectrum auctions. In contrast, our work targets CSS systems and differential location privacy.

Chapter 5

PRICCS: PRIVACY-PRESERVING CROWDSOURCED SPECTRUM SENSING

5.1 Overview

This chapter presents *PriCSS* [79], a novel framework for a spectrum database administrator to select spectrum-sensing participants in a differentially privacy-preserving manner. The specific contributions of this chapter are as follows. First, we formulate participant selection in CSS systems as a reverse auction problem where each participant's true cost for performing the sensing tasks is closely tied with the participant's current location. Second, we demonstrate a location-privacy attack under the previous formulation. Third, we present a new formulation based on the exponential mechanism to offer differential location privacy. Last, we thoroughly evaluate PriCSS through theoretical and simulation studies. Our results confirm that PriCSS can simultaneously achieve the following objectives.

- **Differential location privacy**. PriCSS can prevent any internal or external attacker with arbitrary knowledge from inferring the locations of mobile participants.
- **Approximate social cost minimization**. Social cost is the sum of the real cost of participants completing all the sensing tasks [70]. PriCSS aims to approximately minimize the social cost.
- **Truthfulness**. Each PriCSS participant has no incentive to lie about his sensing cost.

The rest of the chapter is organized as follows. Section 5.2 introduces the system and adversary models. Section 5.3 formulates the participant selection without privacy consideration. Section 5.4 discusses the potential location privacy leakage in the process of candidate participant selection. We show the detailed scheme design of PriCSS in Section 5.5 and provides theoretical analysis in Section 5.6. The evaluation results are demonstrated in Section 5.7. Finally, Section 5.8 concludes our work.

5.2 System and Adversary Models

5.2.1 System Model

PriCSS is run by a spectrum database administrator whose functionalities, however, go far beyond those of the current spectrum database administrators. Specifically, similar to a spectrum database administrator, the PriCSS administrator accepts registrations from primary users and answers the spectrum-occupancy queries from secondary users. In addition, the PriCSS administrator can manage the spectrum of itself or other licensed users by issuing spatiotemporal spectrum permits which allow secondary users to use specific channels at specific locations during specific periods.

The PriCSS administrator relies on mobile crowdsourcing to obtain fine-grained information for its managed spectrum. Crowdsourcing spectrum sensing tasks eliminates the need for the PriCSS administrator to deploy and manage a large-scale sensor network dedicated to spectrum sensing. More specifically, to determine the realtime quality and occupancy of a specific channel in a certain area, the PriCSS administrator recruits mobile users there, referred to as PriCSS participants, to perform spectrum sensing at a set of designated locations. The PriCSS administrator can then make a decision by fusing the sensing reports. This sensing method is known as cooperative spectrum sensing and has been widely studied. The sensing locations

usually should be far apart from each other to ensure high spatial diversity and thus high sensing quality. For the purpose of this chapter, we hereby assume that the PriCSS administrator has pre-determined the sensing locations of each sensing task according to the existing methods such as [80].

Each PriCSS participant is a mobile user who owns an advanced mobile device capable of spectrum sensing. He registers with the PriCSS administrator under his real identity to receive rewards for performing spectrum sensing. Each PriCSS participant also has a unique pseudonym or identifier which is visible to other participants in the system. In contrast, the real identity of each participant is kept confidential to himself or the PriCSS administrator.

5.2.2 Adversary Model

We assume that the PriCSS administrator is fully trusted in preserving the real identity and bids of PriCSS participants. This common assumption can be relaxed by introducing multiple semi-trusted parties who do not collude. How this relaxation can be done is beyond the scope of this chapter.

The adversary can be internal or external to PriCSS. An internal attacker corresponds to a PriCSS participant. We assume that internal attackers are honest-but-curious (HBC) in the sense that they faithfully fulfill promised sensing tasks but have interests in finding out the locations of other PriCSS participants. We also assume that PriCSS participants may lie about their spectrum sensing costs to claim more rewards, but they are rational in the sense that they only lie if they can benefit. Such HBC and rational assumptions are commonly adopted in the literature to model the attackers not performing denial-of-service attacks. In contrast, an external attacker does not participate in PriCSS but tries to infer the locations of PriCSS participants from public information.

We assume that the adversary has arbitrary background knowledge for attempting to breach the location privacy. For example, both internal and external attackers know the details of the system operations, and they may also collude. We intend to offer differential location privacy to each PriCSS participant under this strong adversary model.

As mentioned in Section 4.2, there can be many other security and privacy issues in CSS systems. We resort to the rich literature for effective defenses, e.g., detecting fake sensing results [18, 20, 61, 62, 63, 64] and spectrum misuse [7, 8, 10, 25].

5.3 Participant Selection Without Privacy

We first formulate participant selection in PriCSS as a reverse-auction problem without considering location privacy. For this purpose, we assume that there are totally N PriCSS participants in a large geographic region such as the Los Angeles metropolitan area. Each participant has a unique integer index in $\mathcal{N} = \{1, \cdots, N\}$, which corresponds to his system pseudonym in practice.

We assume that the PriCSS administrator issues M sensing tasks. Each task $m \in [1, M]$ contains one or more channels to sense, a time window in which the sensing should be done, and $\mu_m \geq 1$ sensing locations which are determined by the PriCSS administrator according to existing results such as [80]. Finally, we denote the j-th subtask of task m by $S_{m,j}$, all the μ_m subtasks of task m by $S_m = \{S_{m,j} | j \in [1, \mu_m]\}$, and all the $\sum_{m=1}^{M} \mu_m$ subtasks by $\mathcal{S} = \{S_{m,j} | m \in [1, M], j \in [1, \mu_m]\}$. The participant is allowed to include multiple sensing tasks at once in his sensing bid according his schedule and itinerary. Since all the subtasks for the same sensing task need to be performed in the same (and generally short) time window, we require that each participant can at most perform one subtask for each sensing task.

The cost for spectrum sensing is modeled as follows. The PriCSS administrator publishes a constant factor η to compensate each PriCSS participant for his resource (power, communication, and computation) consumption and human effort incurred for each sensing subtask. Another constant ρ is also published as the travel compensation per unit distance for gas consumption, driving time, etc. For simplicity, we use Euclidean distance to model the travel distance between two points. Assume that a participant chooses to perform q subtasks in a round trip of total Euclidean distance d. His true sensing cost is defined as $v = q\eta + \rho d$. For example, if a participant is currently at position l_1 and wants to perform two subtasks a and b which are located at l_a and l_b, respectively. Then d equals $\text{Euclidean}(l_1, l_a) + \text{Euclidean}(l_a, l_b) + \text{Euclidean}(l_b, l_1)$. Therefore, his true sensing cost for the two subtasks is simply $2\eta + \rho d$. Each participant knows this cost model for computing his sensing cost, and the PriCSS administrator can modify the model based on user feedbacks.

The PriCSS administrator aims to select n_m unique participants for each spectrum sensing task $m \in [1, M]$. Since PriCSS participants compete to perform spectrum sensing tasks in return for rewards, it is reasonable to model participant selection in PriCSS under a reverse combinatorial auction framework [81]. In this framework, the PriCSS administrator serves as an auctioneer to auction the sensing tasks, and each participant $i \in [1, N]$ acts as a bidder for the sensing tasks.

We outline the auction procedure as follows. The PriCSS administrator broadcasts the subtask list $\mathcal{S}$ and expects each interested participant i to reply with one bid $b_i = (Q_i, c_i)$, where $Q_i \subset \mathcal{S}$, and c_i is his claimed cost to perform the sensing subtasks Q_i. We assume that c_i is limited in the range of $[c_{\min}, c_{\max}]$, where $c_{\min}$ and $c_{\max}$ are reasonable minimum and maximum possible sensing cost, respectively. Each participant follows two rules to place his bid. First, he can bid for no more than one subtask for each sensing task. Second, he can bid for multiple sensing tasks. The

first rule is necessary to prevent strategic manipulation of the bids. For example, participants A and B both bid for the same two subtasks $S_{1,1}$ and $S_{2,1}$. If bidding truthfully, A will be allocated with $S_{1,1}$, and B will be allocated $S_{2,1}$. However, A might find out that if he is assigned with $S_{2,1}$, he can gain more rewards. Thus, A could purposely lie about the cost of $S_{1,1}$ to give away the sensing opportunity of $S_{1,1}$ to B. Since B has already been assigned with one subtask for this specific sensing task, B is excluded for consideration of task assignment of $S_{2,1}$. In this way, A purposely lies about one sensing cost to win the other sensing subtask and gains more. Such attacks can be effectively thwarted by the first rule. The second rule is to allow participants to perform multiple spectrum-sensing tasks during a round trip so that the total cost for performing the bundled sensing tasks can be reduced.

Given the bid set $\mathcal{B} = \{b_i | i \in [1, N]\}$, the administrator determines the outcome of the auction, denoted by $\overrightarrow{x}(\mathcal{B}) = \{x_1, x_2, \cdots, x_N\}$, where x_i is an indicator for participant i:

$$x_i = \begin{cases} 1, & i \text{ wins the subtask bundle } Q_i, \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

Correspondingly, the administrator selects a winner set $\mathcal{W}$ such that all subtasks in $\mathcal{S}$ can be fulfilled.

Each participant also holds a true valuation about the performing cost for the subtask set Q_i, which is calculated with the cost model previously and denoted by v_i. The utility of participant i whose bid b_i is accepted is defined as "$u_i = p_i x_i - v_i$," where p_i is the payment the administrator makes to participant i. Note that the utility is normalized to 0 if the participant is not a winner. The participants know the allocation algorithm and the payment scheme in advance, and each participant wants to choose his strategy to maximize his own utility. So the claimed cost c_i might not equal v_i for each participant.

In our model, for each sensing task bundle, the participants could have different valuations due to different sensing and travel costs involved. Since each participant decides his own bundle to bid for, we aim to design a truthful mechanism so that participants have no interests in lying about the claimed cost. In addition, the time interval between consecutive rounds of auctions can be dynamically adjusted by the PriCSS administrator according to its service requirements.

Problem Formulation. We formulate participation selection in PriCSS as follows without considering location privacy.

$$\begin{aligned} \textbf{minimize} \quad & \sum_{i \in \mathcal{W}} c_i \\ \textbf{subject to} \quad & |(\bigcup_{i \in \mathcal{W}} Q_i) \bigcap S_m| = \mu_m, \forall m \in [1, M], \\ & |Q_i \bigcap S_m| \leq 1, \forall m \in [1, M], \forall i \in \mathcal{W}. \\ & |Q_i| \leq \tau, \forall i \in \mathcal{W}. \end{aligned} \tag{5.2}$$

The first condition in the equation above indicates that participants in the winner set can fulfill all the M sensing tasks. The second one requires that each participant bid at most one subtask for each sensing task. The third one is to limit the number of sensing tasks a participant can perform in a single round. τ is a constant and specified by the administrator.

The basic problem can be essentially treated as a minimum weighted set cover problem [82], which is knowingly NP-hard. So our basic problem is also NP-hard, which can be solved by an iterative approximation algorithm as follows. We define the average contributory cost of a participant as his original claimed cost over the number of subtasks which he bids for and are not yet allocated to other participants. In each iteration, the PriCSS administrator selects a new participant who has the minimum contributory cost among the remaining participants. The algorithm terminates when

all the constraints are satisfied. We say that one participant *outbid*s another if the former is chosen earlier than the latter.

5.4 Your Location Is No Secret

In this section, we exemplify some attacks to infer PriCSS participants' locations when they are selected under the reverse auction framework in Section 5.3. The location of a participant here refers to his *base location* (e.g., home or workplace) where he stays for a long time each day, and the base location serves as the reference point for the participant to derive his sensing cost for any interested spectrum sensing tasks. We assume that each participant starts from his base location and returns there after performing spectrum sensing tasks.

We also assume that the PriCSS administrator publicizes each spectrum-sensing auction result to ensure the public that its participant selection is unbiased. The publicized information only includes the system identifier of each participant winning one or multiple sensing subtasks. The real identity, claimed cost, and received payment of each winning participant are still kept confidential. Making the auction result public can also help the winners achieve greater self-esteem and public recognition, for which there are numerous examples in practice. For instance, an Amazon user can get his product reviews seen and voted by others, and those contributing highly voted reviews can get free products to test and keep.

The key insight for the location-inference attacks is that a participant's claimed sensing cost is tied to his round-trip Euclidean distance according to the aforementioned public cost model $v = q\eta + \rho d$, which corresponds to performing m subtasks in a round trip of total Euclidean distance d. Even if the claimed cost of each participant is hidden, the attackers can still infer the locations of some participants from the auction results and the changes in auction participation. We give some attack

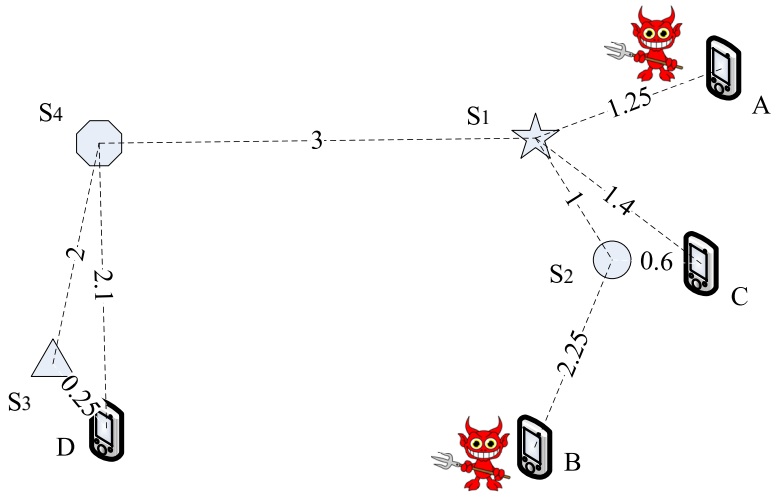


Figure 5.1: A Location-Inference Attack Example.

examples in what follows to highlight the need for preserving location privacy. We consider two rounds of auctions, which involve identical channels and sensing locations but different sensing times. This is practical because the PriCSS administrator may want to know the occupancy and quality of each channel in each service area according to a periodic, on-demand, or random schedule.

Case 1: Single Task.

We first consider a simple case in which each participant can bid for a single sensing task. Since each participant can perform no more than one sensing subtask for any sensing task, the bid of each participant is hence for a single subtask.

For example, consider three participants $\{A, B, C\}$ bidding for the same subtask. According to the aforementioned cost model, their true sensing costs are $v_A = \eta + \rho d_A$, $v_B = \eta + \rho d_B$, and $v_C = \eta + \rho d_C$, respectively, where d_A, d_B, and d_C denote their respective Euclidean distance to the subtask location. Assume that the base locations of A, B, and C do not change. Nor do d_A, d_B, and d_C. In addition, we temporarily assume that the claimed cost of each participant equals his true sensing cost, which can be technically guaranteed later. So we have $c_A = v_A$, $c_B = v_B$, and $c_C = v_C$.

Assuming that $d_A > d_B > d_C$, we have $c_A > c_B > c_C$. According to our formulation in Eq. (5.2), participant C will be selected as the winner in the first round. In the second round (say, next day), assuming that C no longer competes for this subtask for some reason such as work schedule change, so only A and B bid. Then B wins in the second round. The PriCSS administrator publishes the participant selection result in each round.

An external attacker can infer from the public information that $c_A > c_B > c_C$ and hence $d_A > d_B > d_C$, which are something a sensitive user does not want to disclose.

Internal attackers can infer much more information. For example, assume that B is an attacker. Since B knows his own distance d_B and $d_C < d_B$, he can infer that participant C must be inside the *suspicion region*, which is the circle centered at the subtask location with radius d_B. If C additionally participates in other sensing subtasks whose locations are also public, B can draw other suspicion regions for C and infer that C is in the intersected area of the suspicion regions with overwhelming probability. B can also speed up his inference and improve the inference accuracy by colluding with other participants in the PriCSS system.

Case 2: Multiple Tasks.

We also give a more complicated example corresponding to the more general case in Eq. (5.2), in which each participant can bid for multiple subtasks with a single claimed cost. As shown in Fig. 5.1, our example involves four sensing tasks $S_1 \sim S_4$, each involving a single subtask. So we abuse the notations $S_1 \sim S_4$ to denote the four subtasks as well. The number associated with each dotted line in Fig. 5.1 represents the Euclidean distance between the two end locations. Let η be 0.5 and ρ be 1 for the aforementioned cost model $v = q\eta + \rho d$, where m denotes the number of chosen subtasks, and d denotes the round-trip Euclidean distance. The bids submitted by $A \sim D$ are as follows: $b_A = \{\{S_1\}, 3\}, b_B = \{\{S_2\}, 5\}, b_C =$

$\{\{S_1, S_2\}, 4\}, b_D = \{\{S_3, S_4\}, 5.35\}$. According to our formulation in Eq. (5.2), the winner set is $\mathcal{W} = \{C, D\}$. In the second round, assuming that C leaves the area or simply skips the auction, the winner set is $\mathcal{W}' = \{D, A, B\}$. Assume that the PriCSS administrator publishes a re-ordered winner set in each round to conceal each winner's selection order. For example, $\{D, C\}$ and $\{B, D, A\}$ are published as the two rounds' results.

There can be many attack strategies for the above scenario. Due to space limitations, we only discuss one case here, in which A and B collude to infer C's location. The attack involves two steps. First, the attackers need to infer the sensing task bundle that C bids. Second, the attackers estimate the claimed cost of C. The first step can be achieved by studying the difference between the two winner sets, $\mathcal{W}$ and $\mathcal{W}'$. From the attackers' point of view, D's bid must have covered only S_3 and S_4. Otherwise, the winner set would have been changed. It follows that C's bid must have covered at least S_1 and S_2. The remaining question is whether C's bid also covers either or both S_3 and S_4.

There are two possible cases now. In the first case, we assume that D outbids C in the first auction and thus gets S_3 and S_4, so C can only contribute to tasks S_1 and S_2. Since C outbids both A and B, his average contributory cost should be smaller than the smallest of A and B's average contributory cost, which corresponds to $c_C/2 < c_A = 3$ or $c_C < 6$. From Fig. 5.1, the minimum round-trip cost for C to perform S_2, S_1 and S_4 sequentially must be larger than 6 and is incurred when C first visits the S_2 location, then the S_1 location and the S_4 location, and finally C's location. The additional cost is higher if S_3 is involved. So C's bid covers S_1 and S_2 only. Plugging $q = 2$, $\eta = 0.5$, and $\rho = 1$ in the cost model $c_C = q\eta + \rho d_C$, the attackers have $c_C = 1 + d_C$ and thus $d_C < 5$. Since the distance between S_1 and S_2 is 1, the sum of the Euclidean distances from C to S_1 and S_2 is smaller than 4. So

the attackers can infer that C must be inside the ellipse with S_1 and S_2 locations as two foci and the major-axis length equal to 4. C's location can be further narrowed down if additional information is available.

Case 3.

In addition to the two exemplary attacks on location privacy, the participants very close to some subtask locations are likely to have lower claimed costs and higher chances to always win the sensing tasks at those locations, as the aforementioned approximate solution to our formulation in Eq. (5.2) is a deterministic process. Therefore, if a participant appears much more frequently than other participants in repeated auctions for the same sensing subtasks, the attackers can infer that the participant must be very close to one of the subtask locations. This kind of location privacy breach should also be prevented.

5.5 Participant Selection With Differential Location Privacy

Till now we have formulated participant selection in PriCSS as an NP-hard problem and described an approximate solution. We have also demonstrated a few attacks under the basic formulation and solution, which can severely endanger the location privacy of PriCSS participants. In this section, we incorporate differential location privacy into the previous formulation and propose an advanced formulation for participant selection in the PriCSS system to simultaneously achieve approximate social cost minimization, truthfulness, and differential location privacy. In what follows, we first outline some background knowledge to facilitate the presentation and understanding of our scheme. Then we present our advanced formulation with differential location privacy.

5.5.1 Background

Definition 1. An auction is truthful if and only if any bidder's (expected) utility of bidding its true valuation v_i is at least its (expected) utility of bidding any other value c_i [83],

$$u_i(v_i, c_{-i}) \geq u_i(c_i, c_{-i}). \tag{5.3}$$

In the above equation, u_i is the utility of bidder i and c_{-i} is the cost vector for all bidders except i.

Definition 2. A mechanism satisfies the voluntary participation condition if agents who bid truthfully never incur a net loss, i.e., $\text{profit}_i(v_i, (c_{-i}, v_i)) \geq 0$ for all agents i, true value v_i, and other agents' bids c_{-i} [84].

Clearly, the voluntary participation condition is a desired property of our scheme design.

Theorem 7. A decreasing output function admits a truthful payment scheme satisfying voluntary participation if and only if $\int_0^\infty x_i(c_{-i}, u)du \leq \infty$ for all i, c_{-i}. In this case, we can take the payments to be [84]

$$p_i(c_{-i}, c_i) = c_i x_i(c_{-i}, c_i) + \int_{c_i}^{\infty} x_i(c_{-i}, u)du \tag{5.4}$$

Differential privacy is a powerful tool to provide statistical guarantee on the privacy leakage induced by publishing outputs based on sensitive input data sets. The basic idea is that for two almost identical input data sets, the output of the mechanism are nearly identical. The formal definition of differential privacy is as follows [73].

Definition 3. A randomized function $\mathcal{M}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $\mathcal{R} \subseteq Range(\mathcal{M})$,

$$\Pr[\mathcal{M}(D_1) \in \mathcal{R}] \leq \exp(\epsilon) \times \Pr[\mathcal{M}(D_2) \in \mathcal{R}]. \tag{5.5}$$

Approximate differential privacy relaxes on the strict requirement and allows a small additive term in the bound [85].

Definition 4. A randomized function $\mathcal{M}$ gives δ-approximate ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $\mathcal{R} \subseteq Range(\mathcal{M})$,

$$\Pr[\mathcal{M}(D_1) \in \mathcal{R}] \leq \exp(\epsilon) \times \Pr[\mathcal{M}(D_2) \in \mathcal{R}] + \delta. \tag{5.6}$$

The parameter δ ensures that although not all events can satisfy the strong guarantee as specified by Eq. (5.5), the alternation is only for very low probability cases. Hence, it is desired that ϵ and δ to be as close to 0 as possible.

The exponential mechanism is a powerful tool to facilitate mechanism design via differential privacy [74]. The query function defined as $q(A, r)$ maps a pair of the input data set A and candidate outcome r to a real valued "score," with the understanding that the higher score is, the better performance the mechanism can achieve. Specifically, it is defined as

$$\Pr[\varepsilon_q^{\epsilon}(A) = r] \propto \exp(\epsilon q(A, r)). \tag{5.7}$$

The exponential mechanism gives $2\epsilon\Delta$ differential privacy, where Δ is the largest change in q by a single change of the input in A.

The following theorem suggests that the probability of a highly suboptimal output is exponentially low [86].

Theorem 8. The exponential mechanism, when used to select an output $r \in R$, gives $2\epsilon\Delta$-differential privacy, letting R_{OPT} be the subset of R achieving $q(A, r) =$

$\max_r q(A, r)$, ensures that

$$\Pr[q(A, \varepsilon_q^\epsilon(A)) < \max_r q(A, r) - \frac{\ln(|R|/|R_{\mathsf{OPT}}|)}{\epsilon} - \frac{t}{\epsilon}] \leq \exp(-t). \tag{5.8}$$

5.5.2 *Differentially Private Participant Selection*

Due to the NP-hardness of the basic problem, we propose an approximate algorithm, combined with the exponential algorithm, to achieve the desired approximate minimum social cost, low computation complexity, and differential privacy.

The objective of the PriCSS administrator is still to select a set of participants for bundled spectrum-sensing tasks, and we refer to Section 5.3 for the notation. We first define a ranking metric to characterize the administrator's preference for participants, which applies to participant $i \in [1, n]$:

$$r(c_i) = \frac{c_i}{|(\mathcal{S} - Q_{\mathcal{W}}) \bigcap Q_i|}, \tag{5.9}$$

where the set $Q_{\mathcal{W}}$ denotes the set of subtasks included in the current winning bids, i.e., $Q_{\mathcal{W}} = \bigcup_{i \in \mathcal{W}} Q_i$.

The rationale of this definition is as follows. The administrator always tends to select the participant with the lowest claimed cost per subtask that has not yet been included in $Q_{\mathcal{W}}$. In each iteration, each participant's ranking preference is calculated. Then for any remaining participant i who has not be included in the winner list, we adopt the following quality score for the exponential mechanism,

$$q(c_i, x_i) = -r(c_i). \tag{5.10}$$

The "$-$" sign is placed to fit the exponential mechanism in our reverse auction model. It is clear that the smaller $r(c_i)$, the higher the quality score of participant i. This effect is preferred during the winner selection.

Algorithm 1 Participant Selection in PriCSS

Input: Universal set $\mathcal{S}$ of sensing tasks, set $\mathcal{B} = \bigcup_{i \in \mathcal{N}} b_i$ of all submitted bids.

Output: Winner set $\mathcal{W}$, social cost c.

Initialization: $\epsilon' \leftarrow \frac{\epsilon}{\Delta \cdot e \ln(e/\delta)}$, $\mathcal{W} \leftarrow \varnothing$, $c \leftarrow 0$, $Q_{\mathcal{W}} \leftarrow \varnothing$;
while $|\mathcal{S} - Q_{\mathcal{W}}| > 0$ **do**
  for all b_i in $\mathcal{B}$ **do**
    if $Q_i \subseteq Q_{\mathcal{W}}$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} - \{b_i\}$;
    else
      $r(c_i) = \frac{c_i}{|(\mathcal{S} - Q_{\mathcal{W}}) \bigcap Q_i|}$;
    end if
  end for
  for all b_i in $\mathcal{B}$ **do**
    $\Pr[\mathcal{W} \leftarrow \mathcal{W} \bigcup \{i\}] = \frac{\exp(-\epsilon' \cdot r(c_i))}{\sum_{b_j \in \mathcal{B}} \exp(-\epsilon' \cdot r(c_j))}$;
  end for
  Select b_i according to the computed probability distribution.
  if b_i is selected **then**
    $\mathcal{B} \leftarrow \mathcal{B} - \{b_i\}$;
    $\mathcal{W} \leftarrow \mathcal{W} \bigcup \{i\}$;
    $c = c + c_i$;
    $Q_{\mathcal{W}} \leftarrow Q_{\mathcal{W}} \bigcup Q_i$;
  end if
end while
return $\mathcal{W}$, c

The details of the proposed allocation scheme is shown in **Algorithm 1**. According to the exponential mechanism, the probability of participant i being selected as a winner is

$$\Pr(x_i = 1) \propto \exp(-\epsilon' r(c_i)) \ , \tag{5.11}$$

where ϵ' is specified as $\frac{\epsilon}{\Delta \cdot e\ln(e/\delta)}$. Δ is the maximum input difference for c_i, which equals $c_{\max} - c_{\min}$. ϵ and δ are parameters to balance the privacy leakage and efficiency (in terms of social cost minimization in our scenario). Line 11 in **Algorithm 1** can thus be derived considering all the unselected participants. It essentially normalizes the overall participants' selection probability. Based on the selection probability for each remaining participant, participant i is selected as the winner in this iteration. We then remove his bid b_i from $\mathcal{B}$ and include i in the winner set $\mathcal{W}$.

We resort to Theorem 7 for the truthful payment design. Each winner i is paid by the administrator with the amount

$$p_i(c_{-i}, c_i) = c_i x_i(c_{-i}, c_i) + \int_{c_i}^{c_{max}} x_i(c_{-i}, u) du, \tag{5.12}$$

where $x_i(c_{-i}, c_i)$ represents the probability that participant i is selected to perform the sensing task bundle Q_i when i's claimed cost is c_i and others' claimed cost vector is c_{-i}.

5.6 Performance Analysis

In this section, we prove how PriCSS achieves the desired design objectives: differential location privacy, approximate social cost minimization, and truthfulness.

5.6.1 Differential Location Privacy

Theorem 9. For any $\delta \leq 1/2$, PriCSS preserves $((e-1)\epsilon'\Delta\ln(e\delta^{-1}), \delta)$-differential location privacy.

Proof. To facilitate the proof, we first define $\widehat{Q}_i$ as the subtask set that participant i can still contribute to, i.e., $\widehat{Q}_i = (\mathcal{S} - Q_{\mathcal{W}}) \bigcap Q_i$. In two consecutive auction rounds, assume that there are two bidding vectors $\{c_1, c_2, \cdots, c_l, \cdots, c_N\}$ and $\{c'_1, c'_2, \cdots, c'_l, \cdots, c'_N\}$ that differ by only one single element at the lth index. $c_i = c'_i$ for all $i \in [1, N]$ except $i = l$. Differential privacy suggests that with these two bidding vectors as input, the probability that the outputs of the mechanism, i.e., the winner sets $\mathcal{W}$ and $\mathcal{W}'$, are approximately the same. The rationale of our proof is to obtain an exponential upper-bound for $\Pr[\mathcal{W} = \{w_1, w_2, ..., w_p\}]/\Pr[\mathcal{W}' = \{w_1, w_2, ..., w_p\}]$, where $\mathcal{W}$ and $\mathcal{W}'$ are the two ordered winner lists, i.e., w_i is always selected as a winner before w_j for any $j > i$. We give our formal proof below:

$$\begin{aligned}
&\frac{\Pr[\mathcal{W} = \{w_1, w_2, ..., w_p\}]}{\Pr[\mathcal{W}' = \{w_1, w_2, ..., w_p\}]} \\
=&\prod_{i=1}^{p} \frac{\exp(-\epsilon' \cdot c_i/|\widehat{Q}_i|)/\sum_{j\in\mathcal{N}\setminus\pi_i} \exp(-\epsilon' \cdot c_j/|\widehat{Q}_j|)}{\exp(-\epsilon' \cdot c'_i/|\widehat{Q}_i|)/\sum_{j\in\mathcal{N}\setminus\pi_i} \exp(-\epsilon' \cdot c'_j/|\widehat{Q}_j|)} \\
=&\prod_{i=1}^{p} \frac{\exp(-\epsilon' \cdot c_i/|\widehat{Q}_i|)}{\exp(-\epsilon' \cdot c'_i/|\widehat{Q}_i|)} \cdot \prod_{i=1}^{p} \frac{\sum_{j\in\mathcal{N}\setminus\pi_i} \exp(-\epsilon' \cdot c'_j/|\widehat{Q}_j|)}{\sum_{j\in\mathcal{N}\setminus\pi_i} \exp(-\epsilon' \cdot c_j/|\widehat{Q}_j|)} \\
=&\exp(\epsilon' \frac{c'_l - c_l}{|\widehat{Q}_l|}) \prod_{i=1}^{p} \frac{\sum_{j\in\mathcal{N}\setminus\pi_i} \exp(-\epsilon' \cdot c'_j/|\widehat{Q}_j|)}{\sum_{j\in\mathcal{N}\setminus\pi_i} \exp(-\epsilon' \cdot c_j/|\widehat{Q}_j|)},
\end{aligned} \tag{5.13}$$

where $\pi_1 = \emptyset$ and $\pi_i = \{w_1, w_2, ..., w_{i-1}\} (i > 1)$. If $c_l < c'_l$, the second term is smaller than 1. Then

$$\frac{\Pr[\mathcal{W} = \{w_1, w_2, ..., w_p\}]}{\Pr[\mathcal{W}' = \{w_1, w_2, ..., w_p\}]} < \exp(\epsilon' \Delta), \tag{5.14}$$

where Δ is the maximum difference of the bid values for the same set of task bundles.

If $c_l > c'_l$, the first term is smaller than 1. We denote $\alpha_j = c_j - c'_j$, then

$$\begin{aligned}
&\frac{\Pr[\mathcal{W} = \{w_1, w_2, ..., w_p\}]}{\Pr[\mathcal{W}' = \{w_1, w_2, ..., w_p\}]} \\
<&\prod_{i=1}^{p} \frac{\sum_{j \in \mathcal{N} \setminus \pi_i} \exp(-\epsilon' \cdot c'_j / |\widehat{Q}_j|)}{\sum_{j \in \mathcal{N} \setminus \pi_i} \exp(-\epsilon' \cdot c_j / |\widehat{Q}_j|)}. \\
=&\prod_{i=1}^{p} \frac{\sum_{j \in \mathcal{N} \setminus \pi_i} \exp(-\epsilon' \cdot c'_j / |\widehat{Q}_j|)}{\sum_{j \in \mathcal{N} \setminus \pi_i} \exp(-\epsilon' \cdot \alpha_j / |\widehat{Q}_j|) \exp(-\epsilon' \cdot c'_j / |\widehat{Q}_j|)} \\
=&\prod_{i=1}^{p} \mathbb{E}_{j \in \mathcal{N} \setminus \pi_i}[\exp(\epsilon' \cdot \alpha_j / |\widehat{Q}_j|)].
\end{aligned} \tag{5.15}$$

Note that for all $\eta \leq 1$, $e^\eta \leq 1 + (e-1)\eta$. Therefore, for all $\epsilon' \leq 1/\Delta$,

$$\begin{aligned}
&\frac{\Pr[\mathcal{W} = \{w_1, w_2, ..., w_p\}]}{\Pr[\mathcal{W}' = \{w_1, w_2, ..., w_p\}]} \\
\leq&\prod_{i=1}^{p} \mathbb{E}_{j \in \mathcal{N} \setminus \pi_i}(1 + (e-1) \cdot \epsilon' \cdot \alpha_j) \\
\leq&\exp((e-1)\epsilon' \sum_{i=1}^{p} \mathbb{E}_{j \in \mathcal{N} \setminus \pi_i} \alpha_j).
\end{aligned} \tag{5.16}$$

So if $\sum_{i=1}^{p} \mathbb{E}_{j \in \mathcal{N} \setminus \pi_i} \alpha_j$ is upper-bounded, the theorem is established. Based on the proofs in [86], we have $\Pr(\sum_{i=1}^{p} \mathbb{E}_{j \in \mathcal{N} \setminus \pi_i} \alpha_j > \Delta \ln(e\delta^{-1})) \leq \delta$. □

5.6.2 Approximate Social Cost Minimization

Theorem 10. With probability of at least $1 - 1/N^{\mathcal{O}(1)}$, PriCSS can assign spectrum sensing tasks to a set of winners with a social cost of at most $\tau\mathsf{OPT} + \mathcal{O}(\ln N)$, where OPT denotes the optimal (minimum) social cost, and n is the number of participants.

Proof. Let $\mathcal{W}_{\mathsf{OPT}}$ denote the set of winners in the auction with the minimum social cost. We denote an arbitrary set of winners as $\mathcal{W}$ and number the winners according to the order of being selected, i.e., $\mathcal{W} = \{w_1, w_2, ..., w_l\}$.

For each $i \in \mathcal{W}$, we define a set $\mathcal{W}_i$, with the following constraints ($\forall j \in \mathcal{W}_i$):

1. $j \in \mathcal{W}_{\mathsf{OPT}}$;

2. $\widehat{Q}_j \bigcap \widehat{Q}_i \neq \varnothing$;

3. $|\widehat{Q}_j - (\widehat{Q}_j \bigcap \widehat{Q}_i)| = 0$;

4. $\widehat{Q}_j \neq \varnothing$ before i is selected as one winner.

The above constraints suggest that in this arbitrary selection $\mathcal{W}$, the reason that a participant j is not listed is that there is a participant i with a conflicting task set with that of participant j, and i wins. Note that in Eq. (5.8), the q function corresponds to the inverse and unified cost in our scenario. Therefore, by taking $t = \mathcal{O}(\ln N)$, we have

$$-\frac{c_i}{|\widehat{Q}_i|} \geq -\frac{c_j}{|\widehat{Q}_j|} - \mathcal{O}(\ln N) \tag{5.17}$$

with a probability of at least $1 - 1/N^{\mathcal{O}(1)}$.

Since $|\widehat{Q}_j|$ is upper bounded by a constant τ where $\tau \ll N$ when n is large, we have

$$c_j \geq \frac{c_i}{|\widehat{Q}_i|} \cdot |\widehat{Q}_j| - \mathcal{O}(\ln N) \tag{5.18}$$

with a probability of at least $1 - 1/N^{\mathcal{O}(1)}$.

Summing all j ($j \in \mathcal{W}_i$) together, we have

$$\begin{aligned} \sum_{j \in \mathcal{W}_i} c_j \geq & (\frac{c_i}{|\widehat{Q}_i|} - \mathcal{O}(\ln N)) \cdot \sum_{j \in \mathcal{W}_i} c|\widehat{Q}_j| \\ \geq & \frac{c_i}{\tau} - \mathcal{O}(\ln N). \end{aligned} \tag{5.19}$$

with a probability of at least $1 - 1/N^{\mathcal{O}(1)}$. The last step holds because $\sum_{j \in \mathcal{W}_i} |\widehat{Q}_j| \geq 1$.

Summing all $i \in \mathcal{W}$, we have

$$\begin{aligned} \sum_{j \in \mathcal{W}_{\mathsf{OPT}}} c_j = & \sum_{i \in \mathcal{W}} (\sum_{j \in \mathcal{W}_i} c_j + \sum_{j \in \mathcal{W}_{\mathsf{OPT}} \bigcap \mathcal{W}_i} c_j) \\ \geq & \sum_{i \in \mathcal{W}} \frac{c_i}{\tau} - \mathcal{O}(\ln N). \end{aligned} \tag{5.20}$$

This concludes the proof. □

5.6.3 Truthfulness

We finally prove that PriCSS is truthful. Based on Theorem 7, we need to show that the selection of PriCSS is monotone decreasing with an appropriate payment scheme.

Lemma 1. In PriCSS, for each participant i, the probability that i is assigned with the interested spectrum sensing task bundle is monotone decreasing with his claimed cost c_i.

Proof. Due to the randomized property of our scheme, we simply prove that the probability that i is assigned with the interested spectrum sensing task bundle is decreasing when his claimed cost c_i increases in each round of winner selection.

$$\begin{aligned}
&\Pr(\mathcal{W} \leftarrow \mathcal{W} \bigcup \{i\}) \\
=&\frac{\exp(-\epsilon' \cdot r(c_i))}{\sum_{b_j \in \mathcal{B}} \exp(-\epsilon' \cdot r(c_j))} \\
=&\frac{\exp(-\epsilon' \cdot r(c_i))}{\sum_{b_j \in \mathcal{B} \setminus \{c_i\}} \exp(-\epsilon' \cdot r(c_j)) + \exp(-\epsilon' \cdot r(c_i))} \\
=&1 - \frac{\sum_{c_j \in \mathcal{B} \setminus \{c_i\}} \exp(-\epsilon' \cdot r(c_j))}{\sum_{b_j \in \mathcal{B} \setminus \{c_i\}} \exp(-\epsilon' \cdot r(c_j)) + \exp(-\epsilon' \cdot r(c_i))}
\end{aligned} \tag{5.21}$$

In the above equation, if we increase c_i, $r(c_i)$ also increases. Then the exponential term of c_i decreases, causing the overall equation value to decrease. This indicates that if we increase c_i, the probability that $\mathcal{W}$ includes i in every round decreases if i has not been included in previous rounds. □

We thus have the following theorem established.

Theorem 11. PriCSS is truthful.

5.7 Performance Evaluation

In this section, we use simulations to evaluate whether PriCSS can achieve differential location privacy and approximate social cost minimization.

Our simulation setting is as follows. We simulate a square urban area of 1km by 1km. The system administrator issues sensing tasks in response to the queries of secondary users, each with a transmission radius of 300m. The base locations of participants are uniformly distributed, and we vary the number of participants from 300 to 1000 in simulations. In our simulation, the preferred sensing locations are chosen beforehand according to the specific diversity requirement as discussed in Section 5.2. To minimize the overall sensing cost, we want the subtask locations to be as far from each other as possible. We specify a minimum separation distance of 100m for the subtasks in each sensing task. The number of sensing locations (or subtasks) for each sensing task is fixed as 5 in our simulations. We vary the number of sensing tasks K in one round of auction from 3 to 9. Each sensing task is characterized by the locations of the corresponding secondary users, which are uniformly distributed within the region. We also set the modeling parameters $\eta = 100$ reward units and $\rho = 1$ unit per meter. The parameter τ is specified as 3. In addition, we set the bidding cost range $[c_{\min}, c_{\max}]$ to [100,1500]. Note that other configurations of η and ρ lead to similar performance. We omit other cases here due to limited space. The privacy parameter ϵ is chosen as 0.1 or 2 unless otherwise stated, and δ is set to 0.25. The simulations are done in MATLAB, and each result represents the average of 200 runs. Fig. 5.2 shows the social cost distribution for a randomly generated topology with 300 participants. The social cost for each participant is associated with the task bundle he is interested in. We can clearly see that the cost is not uniformly distributed across the range. In addition, the total normalized count does not add

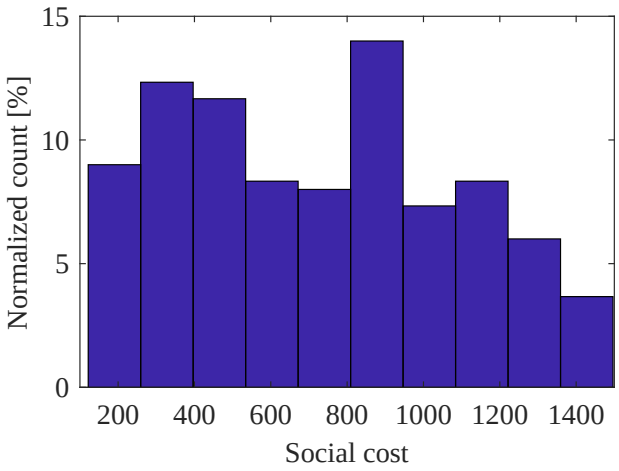


Figure 5.2: Social Cost Distribution for a Randomly Generated Topology With 300 Participants.

up to 1 for the range we show here due to the upper bound of cost (1500) we have imposed in our system. In other words, a small portion of participants are filtered out due to their high social cost.

We use two metrics to evaluate the system performance. The first is the privacy loss, defined according to **Definition** 3:

$$\epsilon = \max_{\mathcal{R}} \ln \frac{\Pr[\mathcal{M}(D_1) \in \mathcal{R}]}{\Pr[\mathcal{M}(D_2) \in \mathcal{R}]}, \tag{5.22}$$

where D_1 and D_2 correspond to two cost vectors for all the participants that differ by one element. Intuitively, the smaller ϵ, the less impact the change of a single cost on the auction results, the better individual sensing-cost privacy is protected, and the more location privacy each participant enjoys. The second metric is the social (or true sensing) cost of the winners for PriCSS, which is desired to be as low as possible. For the purpose of comparison, we also show the social cost induced using the approximation algorithm without privacy considerations introduced in Section 5.3 and use the label "baseline" in the figures.

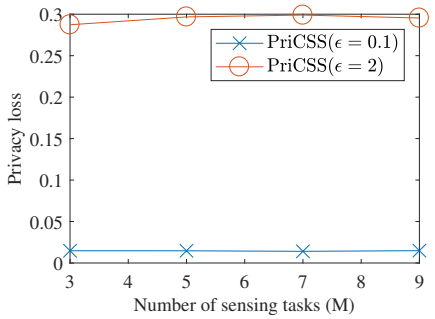
We first evaluate the location-privacy loss in PriCSS. As proved in Section 5.6, PriCSS preserves $((e-1)\epsilon'\Delta\ln(e\delta^{-1}), \delta)$-differential location privacy, where ϵ' is specified as $\frac{\epsilon}{\Delta \cdot e\ln(e/\delta)}$. This is equivalent to achieving $(\frac{e-1}{e}\epsilon, \delta)$ differential privacy. Fig. 5.3a and Fig. 5.3b shows the achievable privacy loss in PriCSS, which is obviously much lower than the theoretical result. Specifically, when $\epsilon = 0.1$, we can observe almost a constant privacy loss of 0.01, which is far lower than the theoretical value $\frac{e-1}{10e} \approx 0.06$. Similar conclusions can be drawn with $\epsilon = 2$. This indicates that when there is any change of a single cost value for any participant, there is rarely any chance that the auction result can change. Hence, we can safely conclude that the attackers can no longer infer the participants' locations by performing the attacks in Section 5.4 or adopting other attack strategies.

We show the social cost incurred using PriCSS and the baseline algorithm for three and nine sensing tasks in Fig. 5.3c and Fig. 5.3d, respectively. For the baseline algorithm, we observe that as the number of participants increases, the social cost tends to decrease due to increased competition among participants. The trend of decrease, however, cannot be found with PriCSS for both $\epsilon = 0.1$ and $\epsilon = 2$ cases. We conjecture that with PriCSS in place, the advantage of cost-efficient participants who claim lower sensing costs in the hope of winning is weakened by the increased number of participants. In other words, their ranking metrics play less significant roles when the number of participants increases. Also, the randomization also introduces more variations. Still, we see that the social cost when $\epsilon = 0.1$ is slightly worse than $\epsilon = 2$. This is the expected trade-off between privacy and utility: the larger ϵ, the heavier weight on the ranking metric, and the lower the social cost. In Fig. 5.3e, we also show the social cost for different numbers of sensing tasks when there are 900 participants. As expected, when the number of sensing tasks increases, the social cost also increases.

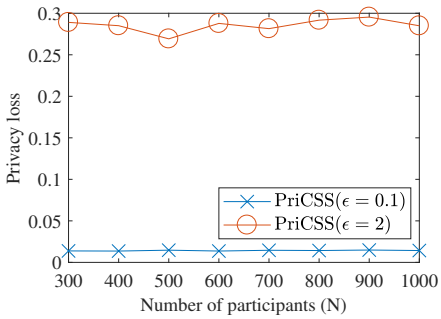
Finally, we show the trade-off between the privacy loss and the total social cost in Fig. 5.3f. As expected, we see that with the increase of ϵ, the privacy loss increases and the total payment decreases.

5.8 Conclusions

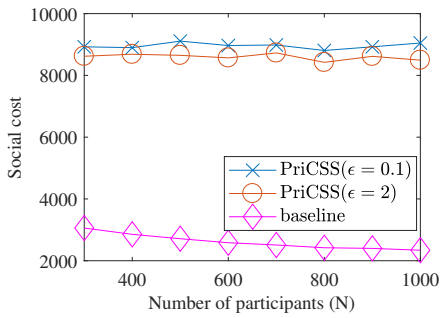
In this chapter, we present PriCSS, a novel framework for a spectrum database administrator to select spectrum-sensing participants in a differentially privacy-preserving manner. We provide detailed privacy and efficiency analysis of the scheme and evaluate the performance extensively. We demonstrate that PriCSS can simultaneously achieve the three design objectives: differential location privacy, approximate social cost minimization, and truthfulness.



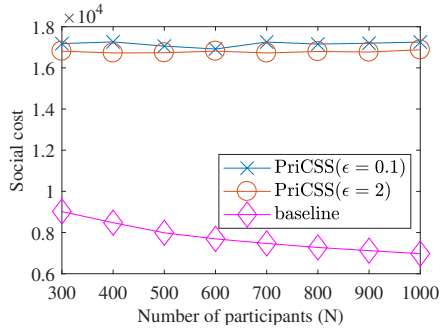
(a) Privacy Loss With 900 Participants



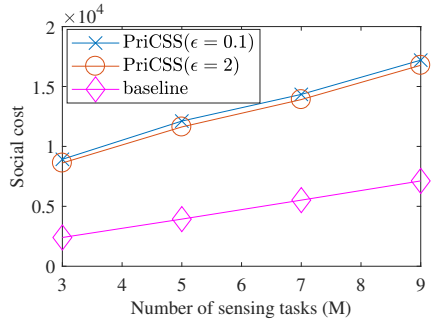
(b) Privacy Loss with 9 Sensing Tasks



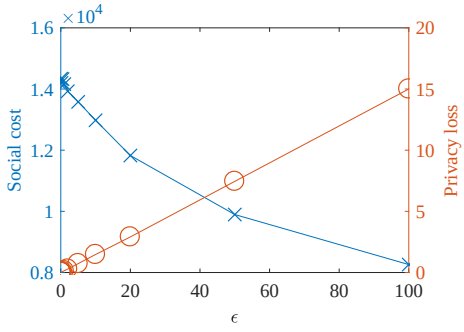
(c) Total Social Cost with 3 Sensing Tasks



(d) Total Social Cost With 9 Sensing Tasks



(e) Total Social Cost With 900 Participants



(f) Trade-Off Between Privacy Loss and Total Social Cost

Figure 5.3: Performance Evaluation for PriCSS.

Chapter 6

DPSENSE: DIFFERENTIALLY PRIVATE CROWDSOURCED SPECTRUM SENSING

6.1 Overview

This chapter presents *DPSense* [87], a novel framework for striking a good balance between location privacy and system efficiency in CSS systems. Different from the previously proposed solution PriCSS, in DPSense, we take a step forward and further assume that the service provider is honest but curious. In DPSense, the SSP publishes spectrum-sensing tasks for specific locations and time periods in the future. Each candidate CSS participant responds to the SSP by submitting his/her predicted (either routine or preplanned) mobility trace which is perturbed to satisfy differential location privacy. We then present an optimization formulation for the SSP to assign spectrum-sensing tasks based on perturbed mobility traces and show that it is NP-hard. Finally, we propose a heuristic solution and thoroughly evaluate it via detailed trace-driven simulations based on real-world mobility traces. Our results confirm that DPSense can simultaneously achieve the following desirable objectives.

- *Differential location privacy.* DPSense offers differential location privacy to mobile participants under a strong adversary model by incorporating the mechanism in [76].

- *Minimal cost or travel distance.* DPSense assigns spectrum-sensing tasks to mobile participants based on their perturbed location traces while ensuring the minimal cost for the SSP or equivalently minimum total travel distance for mobile participants.

- *High task-completion rate.* DPSense guarantees that each spectrum-sensing task can be successfully conducted with overwhelming probability.

The rest of the chapter is structured as follows. Section 6.2 introduces the system and adversary models. Section 6.3 motivates the requirement for location privacy in CSS. Section 6.4 reviews the differential privacy mechanism in [76] underlying DPSense. Section 6.5 presents the DPSense framework. Section 6.6 demonstrates the experimental evaluations. Section 6.7 concludes this chapter.

6.2 System and Adversary Models

In this section, we introduce the system model, the spectrum-sensing model, and the adversary model.

6.2.1 System Model

We consider a CSS system consisting of a spectrum service provider (SSP) and N mobile participants in CSS. In addition to having the similar functionalities to traditional database-driven DSA system operators [48, 22], the SSP explores mobile crowdsourcing to estimate spectrum availability in its service region and answers spectrum access requests from SUs.

Each mobile participant is a user who carries an advanced mobile device with spectrum sensing capabilities and wishes to earn rewards by participating in CSS. The participant registers with the SSP and communicates with the SSP via an app installed on his [1] mobile device. Developed by the SSP, the app is assumed to pass the strict vetting process of the trusted app store and has no unauthorized access to the user's locations.

[1] No gender implication.

The SSP generates a spectrum-sensing task either periodically or on demand upon receiving a spectrum-access request from an SU. Our system works in the same way for both cases. The SSP converts each sensing task into a number of subtasks to ensure that the sensing reports submitted by different mobile participants are independent of each other. In particular, let T_j denote the j-th sensing task, which includes R_j as the physical sensing region, t_j^s as the sensing time period, and div_j as the targeted diversity order to be further explained in Section 6.2.2. The SSP first selects n_j candidate sensing locations in R_j, denoted by $\{l_{j,k}^s\}_{k=1}^{n_j}$, such that any two locations are separated with a distance over d_0, where d_0 is a fixed system parameter. The SSP then generates n_j subtasks $\{S_{j,k}\}_{k=1}^{n_j}$, where $S_{j,k} = (l_{j,k}^s, t_j^s)$. Finally, the SSP assigns subtasks to mobile participants based on their mobility traces. A subtask can be accepted or declined by the chosen mobile participant. Task T_j is said to be *completed* if and only if at least div_j subtasks are accepted by mobile participants.

To enable spectrum-sensing task assignment, each participant i periodically predicts his mobility trace for the upcoming time period and submits it to the SSP. This can be easily done in practice, as most mobile users have target locations to go instead of wandering around. Each mobility trace can be represented as a sequence of ω location and time pairs, $L_i = \langle (t_{i,1}, l_{i,1}), \ldots, (t_{i,\omega}, l_{i,\omega}) \rangle$, where $t_{i,u}$ and $l_{i,u}$ $(\forall u \in [1, \omega])$ denote the u-th time and location points, respectively, and ω is a system parameter. To be more practical, $t_{i,u}$ and $l_{i,u}$ can be the indexes of a time slot and a physical cell, as specified by the SSP. The mobility traces can be either automatically obtained via popular location service APIs such as Google Map API or manually fed to the mobile app by participants. Some participants may opt out of providing their mobility traces, in which case they are considered unavailable for the entire time period.

6.2.2 Spectrum Sensing Model

Each mobile participant performs spectrum sensing by detecting PU transmissions on the specified channel in the time and location designated by the SSP. We adopt the following conventional spectrum-sensing model to facilitate the presentation, but our work can be easily extended to support other sensing models.

We assume that the channels between PUs and mobile participants are Rayleigh fading with additive white Gaussian noise (AWGN). The shadow fading is spatially correlated, and the correlation of the received signals for two spectrum sensors separated by distance d can be modeled as an exponential function $e^{-a\mathsf{d}}$ [80], where a refers to an environment parameter which is approximately 0.1204 and 0.002 in urban non-line-of sight and suburban environments, respectively. The *de-correlation distance* d_0 is defined as the minimum distance for two spectrum sensors when the correlation is under a desired threshold.

We assume that the SSP uses the Neyman-Pearson (NP) detector to combine multiple sensing reports from mobile participants to reliably determine spectrum occupancy. Specifically, for a target average decision error probability P^* that accounts for both false positives and false negatives, the number of independent spectrum-sensing reports needs to be no less than the *diversity order* [88, 37],

$$\mathsf{div}^* = -\lim_{\mathsf{SNR}\to+\infty} \frac{\log P^*}{\log \mathsf{SNR}}, \tag{6.1}$$

where SNR is the average signal-to-noise ratio at the sensing participants. We subsequently assume that the SSP can determine proper div^* for each spectrum-sensing task.

Once the diversity order is concretely defined, the following theorem can be similarly derived according to [88].

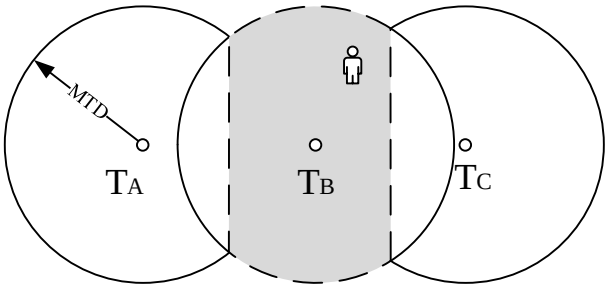


Figure 6.1: An Exemplary Location-Inference Attack, Where the Participant Chooses T_B Over T_A and T_C.

Theorem 12. For multiuser sensing with soft information fusion, when the sensing threshold is chosen to minimize the average error probability, the diversity order of the NP detector equals the number of cooperative users.

Similar conclusions can be drawn for hard decision fusion as well. For details, please refer to [88].

6.2.3 Adversary Model

We assume that the SSP is *honest but curious*, which is commonly used to characterize a reasonable service provider. In particular, the SSP is trusted to faithfully follow the protocol execution but is also interested in learning mobile participants' locations. We assume that the SSP can have arbitrary prior knowledge for attempting to breach the participants' location privacy. In particular, it may infer target mobile participant's location by exploiting the temporal correlation among the submitted mobility traces.

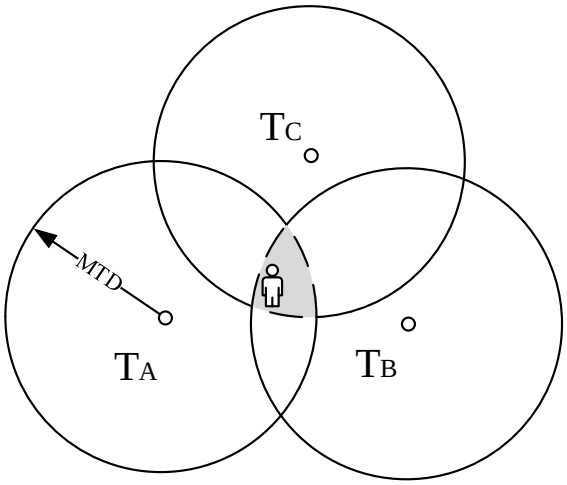


Figure 6.2: Another Exemplary Location-Inference Attack, Where Triangulation Is Used to Locate the Possible Region of the Victim.

6.3 Location Inference in CSS

In the original CSS system, the SSP needs to know the locations of mobile participants for assigning sensing tasks. This requirement obviously violates the location privacy of mobile participants in the desired sensing period. It is worth emphasizing that location privacy here refers to the secrecy of each participant's original mobility trace when he is not involved in CSS. In this section, we illustrate several location-inference attacks against several plausible attempts to improve the location privacy in CSS.

One plausible solution to protecting location privacy in CSS is to let the SSP broadcast spectrum-sensing tasks to all mobile participants who then claim tasks without disclosing their locations to the SSP. Unfortunately, since mobile participants tend to select sensing tasks close to their locations, the SSP could still infer their locations based on the tasks they choose. The reason is that mobile participants

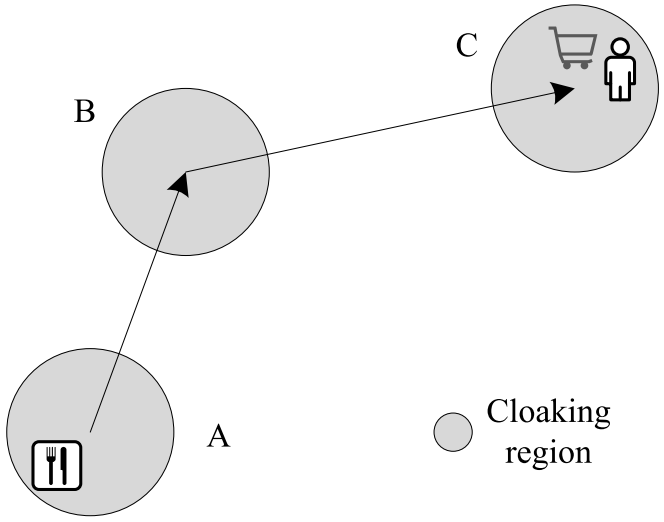


Figure 6.3: Another Exemplary Location-Inference Attack That Explores the Temporal Correlation of Adjacent Reported Locations.

generally are only willing to travel up to a certain distance (e.g., slightly deviating from their scheduled routes), which is commonly referred to as the maximum travel distance (MTD) and can be learned from publicly available data [89]. The sensing task chosen by a mobile participant simply indicates that his location is most likely within the circle centered at the chosen task's center location with a radius of MTD, and such information is what the participants may not want to disclose. The SSP can take one step further to shrink the area a participant resides in from his sensing preference. Consider Fig. 6.1 as an example. Assume that the SSP broadcasted three tasks T_A, T_B, and T_C, where three circles represent their corresponding maximum travel regions. Suppose that a target participant chose task T_B. Under the reasonable

assumption that the participant always chooses the closest task, the SSP can easily confine the participant's location within the shaded area.

A more subtle attack against the above plausible solution is to use trilateration. Assume that the SSP broadcasts one sensing task in one round around the target area but with slight modification of the sensing region, as shown in Fig. 6.2. The three rounds can be carefully scheduled so that during the three rounds, the participant could be very likely located in the same location. For example, the three rounds can be scheduled simply at the same time of the day. In the figure, the participant sequentially chooses the sensing tasks T_A, T_B, and T_C. The SSP could simply use triangulation to find out the intersection of the three regions so that the victim is very likely in the highlighted region. As is shown, the area of the intersection could be very small. Therefore, the participant's location privacy is further compromised.

Another possible solution is to let the participants submit perturbed locations to the SSP which in turn assigns sensing tasks based on perturbed locations. Unfortunately, based on a recent study [76], the SSP can still infer participants' locations by exploiting the temporal correlation among multiple perturbed locations submitted within a short time period. Consider Fig. 6.3 as an example. Suppose that one participant moved from a restaurant in area A to a supermarket in area C and submitted three circular cloaking areas generated from some spatial cloaking mechanism. Although the individual locations were cloaked at each time, the order of the three cloaking areas along with some side information such as road constraints may reveal his exact location at the supermarket.

The three exemplary attacks discussed above highlight the risk of location privacy breach in CSS and call for an advanced solution to protect mobile participants' location privacy.

6.4 Differential Privacy With Temporal Correlation Consideration

In this section, we briefly review the differential privacy mechanism in [76], which DPSense relies on for generating differentially private mobility traces.

6.4.1 Inference Model

We first discuss the Markov chain to model the temporal correlations among the submitted locations of a particular CSS participant. From the SSP's point of view, since it can only observe the perturbed mobility trace instead of the original one, the inference process is a Hidden Markov Model (HMM).

Assume that the sensing region is divided into disjoint cells, indexed from 1 to m_c. Let $p_t = (p_t[1], \ldots, p_t[m_c])$ denote the probability distribution of a certain participant at time t. For example, if a participant at time t is likely to reside in cell 1, 2, 3, and 4 with probability 0.15, 0.25, 0.35, and 0.25, respectively, we have $p_t = \{0.15, 0.25, 0.35, 0.25, 0, ..., 0\}$. Let $M^t = [m_{ij}]$ denote the transition matrix, where m_{ij} is the probability that the participant moves form cell i to cell j for all $1 \leq i, j \leq m_c$ between consecutive timestamps. Given a probability vector p_{t-1}, the probability at time t can be computed as $p_t = p_{t-1}M^t$. We assume that the transition matrix M^t is given as a priori knowledge, which can be generated either from public transportation data or from personal transportation data [2] using existing methods [54]. Since M^t can be constructed based on some public anonymized mobility datasets that are totally unrelated to the participants in our system, it does not negatively affect the location privacy of our system participants.

[2]For example, Google Now continuously tracks users' locations and display relevant information to users in the form of "cards" [90].

We further define the prior and posterior probabilities of a user's location before and after observing the perturbed location at time t as p_t^- and p_t^+, respectively. It is obvious that $p_t^- = p_{t-1}^+ M^t$.

6.4.2 Differential Location Privacy

Differential location privacy is defined over a δ-location set [76].

Definition 5. (δ-Location Set). Let p_t^- be the prior probability of a user's location at time t. The δ-location set is a set containing the minimum number of locations that have the prior probability sum no less than $1 - \delta$:

$$\Delta \mathrm{X}_t = \min\{z | \sum_z p_t^-[z] \geq 1 - \delta\}. \tag{6.2}$$

Definition 6. At any time t, a randomized mechanism $\mathcal{A}$ satisfies ϵ-differential privacy on the δ-location set $\Delta \mathrm{X}_t$ if, for any output $\hat{u}_t$ and any two locations u_1 and u_2 in $\Delta \mathrm{X}_t$, the following holds:

$$\frac{\mathbf{Pr}(\mathcal{A}(u_1) = \hat{u}_t)}{\mathbf{Pr}(\mathcal{A}(u_2) = \hat{u}_t)} \leq e^{\epsilon}. \tag{6.3}$$

To satisfy the differential privacy requirement defined above, a location release algorithm that relies on Markov inference and the planar isotropic mechanism is proposed in [76]. The output of the algorithm is a differentially private version of the input mobility trace. We defer the algorithm outline to Section 6.5.2 for clarity.

6.5 DPSense Framework

In this section, we present the DPSense framework.

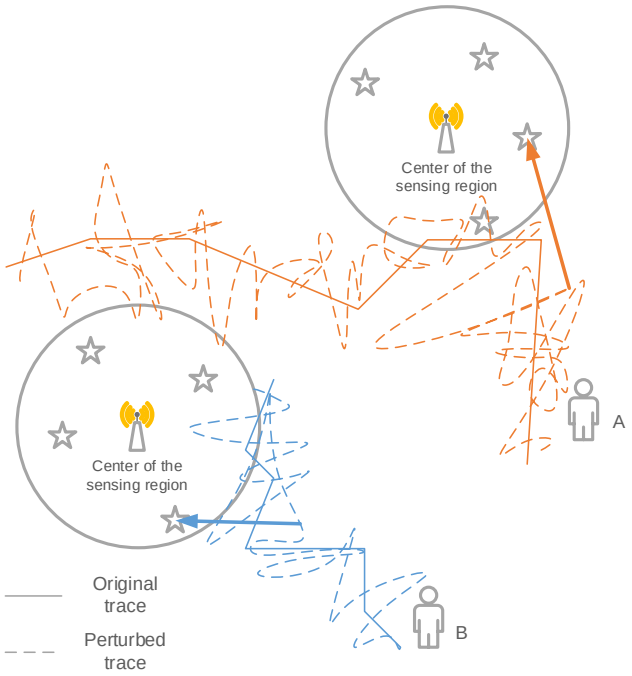


Figure 6.4: The DPSense Framework.

6.5.1 Overview

DPSense is intended to strike a balance between the spectrum-sensing quality, the overall spectrum-sensing cost, and the location privacy. The DPSense framework is illustrated in Fig. 6.4.

Assume that the SSP has M sensing tasks, denoted by $\mathcal{T} = \{T_j\}_{j=1}^{M}$, to fulfil in a future time period, e.g., starting one hour later. Each mobile participant i submits his predicated mobility trace either periodically or in response to the SSP's request. Recall that the mobility trace of participant i in the target sensing period is defined as $L_i = \langle (t_{i,1}, l_{i,1}), \ldots, (t_{i,\omega}, l_{i,\omega}) \rangle$. Instead of submitting L_i to the SSP, participant i submits a perturbed version L_i^o based on the algorithm in [76]. Subsequently, the SSP smooths the perturbed traces according to the procedure in Section 6.5.3 and finally

runs our proposed algorithm in Section 6.5.6 on the smoothed mobility traces to assign the M sensing tasks. As discussed, each task can be divided into a number of subtasks at different locations in the desired sensing region. Each mobile participant receives either zero or one subtask assignment, and he may accept or decline the assignment (e.g., when the subtask location is too far from his original route). A sensing task is completed if the number of mobile participants accepting the subtask assignment is no less than the predefined diversity order. Each participant could be granted some monetary rewards or reputation points that are proportional to the distance he has to travel to perform the sensing task. How the participants are actually rewarded is orthogonal to this chapter.

6.5.2 Generating Differentially Private Mobility Traces

We use the location release algorithm in [76] which is based on Markov inference and the planar isotropic mechanism (PIM). The algorithm accepts a true mobility trace as input and outputs a perturbed mobile trace that satisfies differential privacy on the δ-location set. Specifically, the algorithm sequentially perturbs each location in the mobility trace through the following steps.

- First, prior probabilities are derived using posterior probabilities and the matrix M^t based on the Markov model.
- Second, a δ-location set is generated to identify the set containing the minimum number of locations that have prior probability sum no less than $1 - \delta$.
- Third, the location at the current timestamp is perturbed by adding a noise generated using PIM based on the K-norm mechanism.

- Fourth, location inference is conducted based on the output perturbed location to update the posterior probability of the user in each location of the δ-location set.

It is proved in [76] that the algorithm guarantees differential privacy. We subsequently call the perturbed mobility trace as the PIM trace and refer interested readers to [76] for detailed illustrations.

6.5.3 Smoothing Perturbed Mobile Traces

Since the SSP can only use the perturbed mobility trace for task assignment, it is intuitive that the closer the perturbed trace is to the original location trace, the more accurate the SSP can estimate each participant's travel cost, and the higher probability that the sensing task can be completed while ensuring differential location privacy to mobile participants. It is therefore essential to reduce the negative impact the noise added to the mobility trace. Recall that for the original location at each timestamp, noise is generated in the isotropic space using the K-norm mechanism. The probability of generating noise of a certain value and the probability of generating noise of the exact inverse value are the same. By averaging multiple consecutive locations, the deviation of the averaged location to the original true location could be smaller in contrast to the difference between the disturbed location to the original true location. When the noise amplitude is large, the average could reduce the negative impact introduced due to the noise.

Based on the above intuition, we propose to smooth each user's differentially private mobility trace using a sliding window and assign tasks based on smoothed location traces. Specifically, we define the size of the sliding window as μ, where μ is an odd integer and system parameter. For each timestamp, we generate a smoothed location as the average of the previous consecutive $\lfloor \mu/2 \rfloor$ PIM locations, the current

Algorithm 2 PIM Traces Smoothing

Input: A set of PIM traces $\{L_i^o\}_{i=1}^N$ and sliding window size μ.

Output: A set of smoothed traces $\{L_i^h\}_{i=1}^N$.

1: **for all** $i \in \{1, \ldots, N\}$ **do**

2: $L_i^h \leftarrow \emptyset$.

3: **for all** $\kappa \in \{\lfloor \mu/2 \rfloor + 1, \ldots, \omega - \lfloor \mu/2 \rfloor\}$ **do**

4: $l_{i,\kappa}^h = \frac{1}{\mu} \sum_{x=\kappa-\lfloor \mu/2 \rfloor}^{\kappa+\lfloor \mu/2 \rfloor} l_{i,x}^o$

5: $L_i^h \leftarrow L_i^h \bigcup \{(l_{i,\kappa}^h, t_{i,\kappa})\}$

6: **end for**

7: **end for**

8: **return** $\{L_i^h\}_{i=1}^N$.

PIM location, and the next consecutive $\lfloor \mu/2 \rfloor$ PIM locations. The details of the smoothing algorithm are summarized in Algorithm 2. We will show in our simulations the effectiveness of the sliding window and the impact of μ.

6.5.4 Accepting/Declining Task Assignments

Participants may accept or decline an assigned sensing task for various reasons. We now introduce a model to characterize the probability that an assigned task is accepted, which takes into account of both the physical travel distance and potential wait time.

We first consider the impact of physical travel distance. According to our system model in Section 6.2.1, each task T_j includes R_j as the physical sensing region, t_j^s as the sensing time period, and div_j^* as the targeted diversity order. The SSP further divides T_j into n_j subtasks $\{S_{j,k}\}_{k=1}^{n_j}$ at locations $\{l_{j,k}^s\}_{k=1}^{n_j}$, respectively. Consider subtask $S_{j,k}$ and participant i as an example. Let L_i be participant i's true mobility

trace and v_{avg} be the average speed. For participant i to travel from location $l_{i,\kappa}$ at time $t_{i,\kappa}$ to perform subtask $S_{j,k}$ at sensing location $l^s_{j,k}$, the time of arrival at the sensing location is subject to the following condition,

$$\mathsf{dist}(l_{i,\kappa}, l^s_{j,k}) \leq v_{avg}(t^s_j - t_{i,\kappa}), \tag{6.4}$$

where $\mathsf{dist}(\cdot,\cdot)$ denotes the Euclidian distance.

We then consider the participant's potential waiting time. In particular, participant i may arrive at the sensing location $l^s_{j,k}$ early. If he needs to wait for a long time period to perform the task, he may reject the task at the very beginning. We therefore define *synthetic distance* to jointly consider the travel distance and the waiting time for a given sensing task, which is computed as

$$\begin{aligned}
&\mathsf{dist}^*(l_{i,\kappa}, l^s_{j,k}) \\
&= \begin{cases}
\mathsf{dist}(l_{i,\kappa}, l^s_{j,k}) + & \\
\alpha v_{avg}(t^s_j - t_{i,\kappa}) - & \text{if } \mathsf{dist}(l_{i,\kappa}, l^s_{j,k}) \leq v_{avg}(t^s_j - t_{i,\kappa}), \\
\alpha \mathsf{dist}(l_{i,\kappa}, l^s_{j,k}) & \\
\infty & \text{otherwise.}
\end{cases}
\end{aligned} \tag{6.5}$$

Synthetic distance defined above essentially converts the waiting time into additional travel distance. The system parameter α indicates the weight of the waiting-time equivalent distance versus that of the true travel distance. Since simply waiting generally involves less effort in comparison with the actual travel, it is reasonable to require that $\alpha \leq 1$.

We use a simple linear distribution model to characterize the probability that participant i will accept subtask $S_{j,k}$. Let the MTD be the maximal travel distance within which a participant is willing to travel to perform a sensing task, which can be obtained from historical data [89]. Similar to [56], we calculate the probability that

participant i will accept subtask $S_{j,k}$ at sensing location $l^s_{j,k}$ by departing from $l_{i,\kappa}$ at time $t_{i,\kappa}$ as

$$\begin{aligned}&\mathbf{Pr}[P_i \leftarrow S_{j,k}|t_{i,\kappa}] \\ &= \begin{cases} 1 - \frac{\mathsf{dist}^*(l_{i,\kappa}, l^s_{j,k})}{\text{MTD}} & \text{if } \mathsf{dist}^*(l_{i,\kappa}, l^s_{j,k}) < \text{MTD}, \\ 0 & \text{otherwise.} \end{cases}\end{aligned} \tag{6.6}$$

In other words, $\mathbf{Pr}[P_i \leftarrow S_{j,k}|t_{i,\kappa}]$ is one when the corresponding synthetic distance is zero and zero when the synthetic distance exceeds MTD.

6.5.5 Spectrum-Sensing Task Assignment Formulation

We formulate the spectrum-sensing task assignment as an optimization problem as follows.

We define the indicator variable $b^{i,\kappa}_{j,k}$ such that $b^{i,\kappa}_{j,k} = 1$ if participant i is assigned to depart from his location $l_{i,\kappa}$ at time $t_{i,\kappa}$ to perform sensing subtask $S_{j,k}$ at sensing location $l^s_{j,k}$ and 0 otherwise. If $b^{i,\kappa}_{j,k} = 1$, the expected diversity order contributed by participant i can be computed as

$$\mathsf{div}^{i,\kappa}_{j,k} = \mathbf{Pr}[P_i \leftarrow S_{i,j}|t_{i,j}], \tag{6.7}$$

where $\mathbf{Pr}[P_i \leftarrow S_{i,j}|t_{i,j}]$ is given in Eq. (6.6).

Given the set of N participants with smoothed PIM traces $\{L^h_i\}^N_{i=1}$ and the set of sensing subtasks $\{S_{j,k}|1 \leq j \leq M, 1 \leq k \leq n_j\}$, we formulate the task assignment as

an integer programming problem as follows.

$$
\begin{aligned}
\textbf{minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\omega} b_{j,k}^{i,\kappa} \cdot \mathsf{dist}^*(l_{i,\kappa}^h, l_{j,k}^s) \\
\textbf{subject to} \quad & \sum_{i=1}^{N}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\omega} b_{j,k}^{i,\kappa} \cdot \mathsf{div}_{j,k}^{i,\kappa} \geq \beta \cdot \mathsf{div}_j^*, \forall 1 \leq j \leq M, \\
& \sum_{i=1}^{N}\sum_{\kappa=1}^{\omega} b_{j,k}^{i,\kappa} \leq 1, \forall 1 \leq j \leq M, 1 \leq k \leq n_j, \\
& \sum_{j=1}^{M}\sum_{k=1}^{n_j}\sum_{\kappa=1}^{\omega} b_{j,k}^{i,\kappa} \leq 1, \forall 1 \leq i \leq N, \\
& b_{j,k}^{i,\kappa} \cdot \mathsf{dist}(l_{i,\kappa}^h, l_{j,k}^s) \leq v_{avg}(t_j^s - t_{i,\kappa}), \\
& \forall 1 \leq i \leq N, 1 \leq j \leq M, 1 \leq k \leq n_j, 1 \leq \kappa \leq \omega, \\
& b_{j,k}^{i,\kappa} \in \{0, 1\}, \\
& \forall 1 \leq i \leq N, 1 \leq j \leq M, 1 \leq k \leq n_j, 1 \leq \kappa \leq \omega.
\end{aligned}
\tag{6.8}
$$

where $\mathsf{dist}^*(\cdot,\cdot)$ denotes the synthetic travel distance. β is a ratio equal to or larger than 1, indicating the importance that the SSP can achieve the desired diversity order for every sensing task. A larger β value generally guarantees that the desired diversity order can be achieved, but some tasks can be over-fulfilled, leading to a higher cost. In contrast, with a smaller β value such as 1, it is most likely that only some sensing tasks can achieve the desired diversity order while others are under-fulfilled. This may not be desired because the lack of diversity in spectrum sensing can lead to false decisions on spectrum availability, leading to potential interference with PU transmissions. We will fully evaluate the impact of parameter β in this chapter. In the above formulation, the first constraint means that the sum of expected diversity order should be no less than the the diversity order required for each sensing task. The second constraint indicates that every subtask can be assigned to at most one participant. The third constraint means that every participant can be assigned at most one subtask. The fourth constraint means that if participant i is assigned to

leave at time $t_{i,\kappa}$ from location $l_{i,\kappa}$ at speed v_{avg} to fulfill subtask $S_{j,k}$, he must arrive no later than t_j^s. Here we assume that each participant can at most complete one sensing subtask for the specified time period. How to assign multiple sensing subtasks to the same participant for sequential completion is left as our future work.

The above problem can be proved to be NP-hard by reducing it to the k-partial set cover problem, which is a generalization of the well studied set cover problem.

Definition 7. (k-partial Set Cover [91]) Given a set $\mathcal{B} = \{b_1, b_2, ..., b_n\}$, a collection $\mathcal{S}$ of subsets of $\mathcal{B}$, $\mathcal{S} = \{S_1, S_2, ..., S_m\}$, a cost function c: $\mathcal{S} \rightarrow \mathcal{Q}^+$, and an integer k, find a minimum cost sub-collection of $\mathcal{S}$ that covers at least k elements of $\mathcal{B}$.

Theorem 13. The integer programming problem defined in Eq. 6.8 is NP-hard.

Proof. We first take a look at a special problem derived from the formulation in Eq. 6.8, where β is 1, and $\mathsf{div}_{j,k}^{i,\kappa}$ is 1 for all valid i, κ, j, k values. The problem derived from Eq. 6.8 involves a series of timestamps to consider. To simplify the analysis, we first focus on a single timestamp $\kappa \in [1, \omega]$. For this single timestamp, participants can only contribute a sensing diversity gain of value 1 if they meet the fourth constraint in sensing time. Hence, we can incorporate the time constraint into participant i's new travel distance $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$. When participants satisfy the fourth constraint, we let $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$ equal $\mathsf{dist}^*(l_{i,\kappa}^h, l_{j,k}^s)$. Otherwise, $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$ is ∞. Then we set to obtain the minimum $\widetilde{\mathsf{dist}}(l_{i,\kappa}^h, l_{j,k}^s)$ for each participant among all the timestamps in $[1, \omega]$, and we denote this value by $\widetilde{\mathsf{dist}}(l_{i,\kappa_{o_i}}^h, l_{j,k}^s)$, where κ_{o_i} is the best timestamp for participant i to leave for sensing location $l_{j,k}^s$ to achieve the lowest travel cost. Hence, the optimization objective can be changed to the minimization of $\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{n_j} b_{j,k}^{i,\kappa_{o_i}} \cdot \widetilde{\mathsf{dist}}(l_{i,\kappa_{o_i}}^h, l_{j,k}^s)$. So now if we focus on a single sensing task $j \in [1, M]$, the problem has already been reduced to the k-partial set cover problem as defined above. In the definition, $\mathcal{B}$ corresponds to the subtask set $\{S_{j,k} | 1 \leq k \leq n_j\}$.

The collection of $\mathcal{S}$ corresponds to the task assignment: participant i to fulfill subtask $S_{j,k}$ for all i and k. The cost function c maps $\mathcal{S}$ to $\mathcal{B}$ by the cost we defined using $\widetilde{\mathsf{dist}}(l^h_{i,\kappa_{o_i}}, l^s_{j,k})$. k is the diversity order constraint div^*_j. So the optimization problem is now a k-partial set cover problem. We then need to solve this problem for all $j \in [1, M]$.

Since the special problem is NP-hard, we now conclude that the original problem defined in Eq. 6.8 is NP-hard. □

Note that there are alternative ways to formulate the optimization. For example, it is possible to minimize the expected total synthetic travel distance or the maximum synthetic travel distance. These alternatives are left as future work.

6.5.6 A Heuristic Solution

We now introduce a heuristic approach to assign subtasks to participants based on their smoothed location traces.

The overall assignment process is summarized in Algorithm 3. The intuition is to sequentially assign every subtask of one sensing task to each participant with the smallest synthetic travel distance until the total expected diversity order exceeds the required threshold. Specifically, the algorithm takes the sensing tasks $\mathcal{T}$, subtask set $\{S_{j,k}\}_{1\leq j\leq N, 1\leq k\leq n_j}$, participant set $\mathcal{P}$, and PIM trace set $\{L^o_i\}^N_{i=1}$ as input and then outputs all subtask assignments. Line 1 smooths all the PIM traces using Algorithm 1. Lines 2 to 10 compute the synthetic travel distance for every participant with every possible departing location and every subtask $\{S_{j,k}\}^{n_j}_{k=1}$. The WHILE loop in Lines 12 to 19 assigns one subtask to one participant, whose synthetic travel distance is the smallest among all. The WHILE loop terminates when the accumulative expected diversity order exceeds the diversity order required for the sensing task T_j.

6.5.7 Participant Response

The SSP informs every selected participant about the subtask he is assigned to. On receiving the subtask assignment, each participant calculates the true physical and synthetic travel distance using his true predicted locations and then informs the SSP whether he accepts the assignment based on the task acceptance model in Section 6.5.4. If the participant agrees to fulfill a certain task, he will need to be at the sensing location in the specified time to perform spectrum sensing. Since the participants win the opportunity to perform the task based on the expected mobility traces, the payments or rewards made by the SSP to the participants should be proportional to the travel distances calculated using the expected mobility traces as well. It is possible that the expected mobility traces provided by the participants differ from the real mobility traces. In such cases, participants still need to make sure that they can perform spectrum sensing in a timely manner. The SSP can set up various types of mechanisms to handle the cases when participants fail to fulfill the sensing tasks they previously agreed to fulfill. For example, a reputation system can be constructed to model the reliability of each participant. When participants fail to perform certain tasks, their reputations in the system decrease, and so do their payments received for performing the sensing tasks. In addition, since participants' failure to perform sensing tasks could possibly lead to unsatisfied diversity requirement, the SSP could assign a discounted diversity gain when certain participants with bad history are selected. How to design a fully workable reputation system remains as our future work.

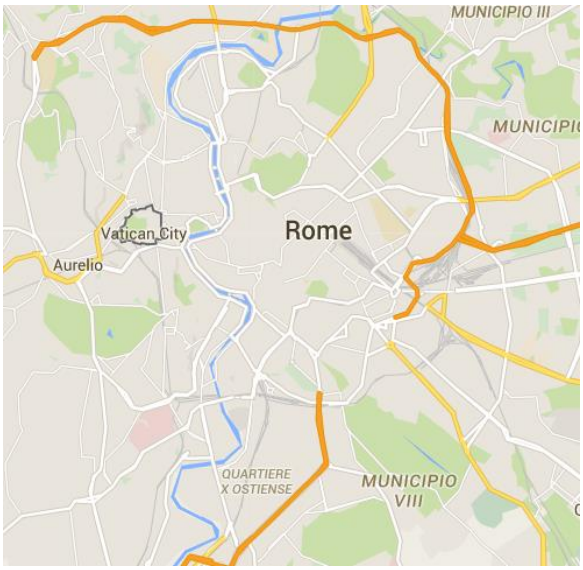


Figure 6.5: The City Area Where the Mobility Traces Are Extracted.

6.6 Simulation Results

In this section, we present the experimental evaluation results of DPSense. We adopt the knowledge construction module in [54] to build the Markov transition matrix, which is implemented in C++. All other modules are implemented in MATLAB on a PC with 2.67 GHz Intel i7 CPU and 9 GB memory.

6.6.1 Mobility Trace Dataset

We use the CRAWDAD dataset roma/taxi [3, 4] for our simulations. The dataset contains the mobility traces of approximately 320 taxis collected over 30 days in Rome, Italy. Each mobility trace consists of a sequence of GPS coordinates collected roughly every seven seconds along with corresponding timestamps. In addition, the taxis in the dataset are not always moving at a high speed. Those idling at one location or moving within a small region can be used to simulate the static participants or the

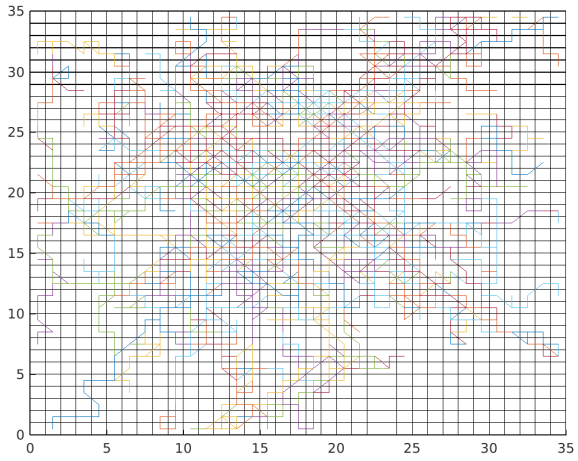


Figure 6.6: Sampled Taxi Mobility Traces From Dataset [3, 4].

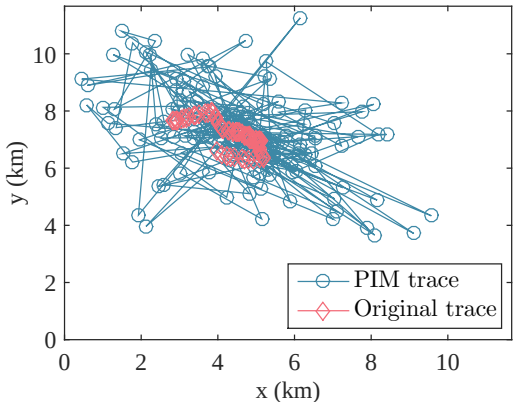


Figure 6.7: The Original Trace and the PIM Trace ($\epsilon = 1$).

participants with very limited moving regions. In our simulations, the time difference between two consecutive timestamp is 20 seconds.

The mobility traces within the center of Rome city are extracted. We consider an area of 11.66 $\times$ 11.66 [km $\times$ km] as illustrated in Fig. 6.5. We divide the area into a 35 $\times$ 35 grids of equal size. We then extract 2700 mobility traces in total, each of which contains 150 timestamps. The 2700 mobility traces are shown in Fig. 6.6. We quantize each GPS coordinate by mapping them into one of the 35 $\times$ 35 cells. As

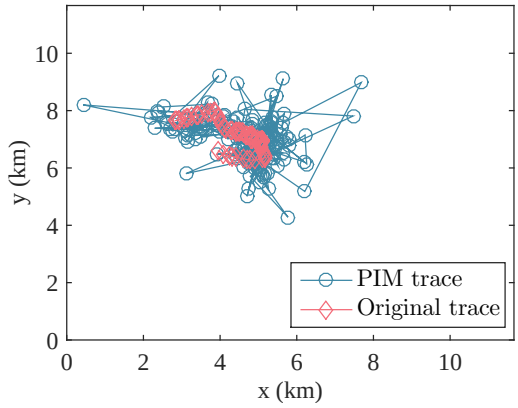


Figure 6.8: The Original Trace and the PIM Trace ($\epsilon = 2$).

we can see, most of the traces are clustered in the center area, resulting in a very dynamic and diverse transition. The density of mobility traces in the four corners is much lower than that in the center area, making it challenging to correctly track the true locations using the PIM scheme. Out of the 2700 mobility traces, 2000 are used to build the Markov transition matrix, and the remaining 700 are used to represent the participants' input traces in our system. The division of the mobility trace dataset is to emulate the practical application scenarios where the SSP can only obtain the historical mobility data based on some large-scale generic location traces which can be completely independent of the participants of our system. Therefore, the construction of the transition matrix does not adversely affect participants' location privacy.

6.6.2 Simulation Setting

We consider a time period of 50 minutes. The sensing tasks are all scheduled at the later half of the 50 minutes because it takes time for participants to arrive at the designated sensing locations. Since the timestamps are in the granularity of 20 seconds, each sensing task is scheduled at a random one of the last 75 timestamps.

We set the simulation parameters as follows. The numbers in bold are the default values if not mentioned otherwise. For the generation of PIM traces, ϵ is chosen among [**1**,2,3,4], and δ is in the range of [0.01,**0.02**,0.03,0.04]. The number of participants N is [400,**500**,600,700]. For the spectrum-sensing task assignment, the size of the sliding window μ is chosen from [1,3,5,**7**,9], where $\mu = 1$ corresponds to the case where no sliding window is used. In addition, we assume that all the participants have the same traveling speed v_{avg}=30 km/h. We expect that higher moving speed will deliver better results because participants can travel a longer distance. Other parameters are set as follows. The maximum travel distance MTD is 15km. The number of sensing tasks M is chosen from [4, **6**, 8, 10] with the number of subtasks $n_j = 10$ for every sensing tasks. The minimum separation distance between sensing locations d_0 is 20m. The sensing region for every sensing task is a circle with radius $R = 300m$. The sensing tasks are randomly generated with the diversity order requirement div^* chosen from [**4**, 5, 6, 7]. The system parameter α is in the range of [0.8, 0.9, **1**], and the parameter β is chosen from [**1**, 1.2, 1.4, 1.6].

In our results, each data point represents the average of 100 runs. We use ΔX to represent the δ-location set. We also compare DPSense with the baseline scheme which does not consider location privacy and use raw mobility traces.

6.6.3 Performance Metrics

For the generation of PIM traces, we compare the *distance error* (i.e., the Euclidean distance between the original trace and the PIM trace for every timestamp) and $|\Delta X|$ (the size of the δ-location set). We also use the following metrics for performance evaluation.

- *Total travel distance (TTD).* This refers to the sum of the expected synthetic distances of the participants who accept assigned subtasks. Specifically, let

$c^i_{j,k}$ be the indicator variable such that $c^i_{j,k} = 1$ if participant i accepts the assigned subtask $S_{j,k}$ and zero otherwise. To achieve the minimum synthetic cost, participant i needs to leave at κ^*th timestamp for the sensing location $l^s_{j,k}$. TTD is then computed as

$$\text{TTD} = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{n_j} c^{i,\kappa^*}_{j,k} \cdot \mathsf{dist}^*(l_{i,\kappa}, l^s_{j,k}). \tag{6.9}$$

TTD is commensurate with the total cost of the SSP for performing all the sensing tasks.

- *Task completion rate (TCR).* TCR is the ratio between the number of tasks that meet the specified diversity order requirements and M (the total number of sensing tasks).

We consider the performance comparison of TTD for three cases: the baseline scheme using the original trace $L_i, \forall i \in [1, N]$; the smoothed PIM trace L^h_i, $\forall i \in [1, N]$; the worst case. The worst case assumes that no chosen participant rejects the assigned subtask. Mathematically, the worst-case TTD is defined as

$$\text{TTD}_w = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{n_j} \sum_{\kappa=1}^{\omega} b^{i,\kappa}_{j,k} \cdot \mathsf{dist}^*(l_{i,\kappa^*_i}, l^s_{j,k}), \tag{6.10}$$

where κ^*_i is the timestamp when participant i leaves for subtask $S_{j,k}$ with minimum synthetic distance.

6.6.4 PIM Trace Generation

We evaluate the impact on PIM trace generation by adapting ϵ values. Since the scheme in [76] is adopted to generate the PIM traces, we refer interested readers to [76] for detailed analysis and evaluations. Per our simulations, we find that δ cannot be too large or too small. With a large δ, locations with small prior probabilities are

likely to be excluded in the δ-location set, ΔX. This is good in keeping a reasonable size of ΔX. However, it might fail to track location updates as well. On the other hand, with a small δ, more locations with small prior probabilities are likely to be included in ΔX. This might lead to a large ΔX (over 40 or more) and result in failure of correctly tracking the true location. Here, we choose $\delta = 0.02$ to achieve a good trade-off. We will present the impact of δ on our system performance later in Section 6.6.9.

We first extract one random participant and visually examine the PIM trace generated in Fig. 6.7 and Fig. 6.8 with different ϵ. It is obvious that $\epsilon = 2$ generates a PIM trace closer to the original trace, which indicates small location errors. This is further confirmed in Fig. 6.9a and Fig. 6.9b. In these two figures, we compare the distance between the original trace and the PIM trace for every timestamp. It is clear that when ϵ is 1, the distance sometimes can be very high (e.g., over 5 km). When ϵ increases to 2, the max distance error is around 3.5 km, which is a huge improvement. We further apply the sliding window in Fig. 6.9c and Fig. 6.9d. We can see that the sliding window is very effective in reducing the distance errors. For example, when ϵ is 1, the max distance error is reduced from 5.1 km to 3.2 km. The average distance error is greatly reduced as well. Reducing the distance error can greatly benefit TCR because the SSP has more accurate knowledge about the participants' locations. Lastly, we compare the size of ΔX in Fig. 6.9e and Fig. 6.9f. We can see that the ΔX size when ϵ is 1 is larger than that when ϵ is 2. This is expected since with a larger ϵ, the mechanism is supposed to track the true location better. We will further evaluate the relationship between ΔX size and ϵ later.

6.6.5 *Effectiveness of Sliding Window*

Fig. 6.12a shows the impact of μ on the total travel distance (TTD). We can see that a larger μ generally results in a smaller TTD. We note that when μ is 1, which corresponds to the case where no smoothing is conducted, with the sliding window in place, TTD has been reduced from 67 km to 48 km, a 28.3% reduction. On the other hand, TTD slows down the reduction when μ is over 5. However, there is still a gap between our scheme and the baseline scheme. This is the expected utility trade-off when incorporating privacy protection. We also show the worst case for comparison. Recall that the worst case is defined as the case that no participant declines the assigned subtask. In other words, SSP simply assigns the sensing task to participants who are the closest to the sensing locations. We see that there is a gap between the worst case and the PIM trace curve. Fig. 6.12b shows the TCR performance. Here, we see that the TCR is dramatically improved from 0.7 to 0.9. Still, TCR slows increasing when μ passes 5. Generally, the larger μ is, the better performance we can achieve in both TTD and TCR. On the other hand, larger μ means higher computation complexity.

6.6.6 *Impact of N*

Fig. 6.12c shows the impact of N, the number of participants on TTD. Generally, the larger N is, the smaller TTD is. This is true for both schemes. Fig. 6.12d shows the TCR performance. Clearly, more participants can lead to higher TCRs. We show the comparison with the baseline scheme using the original trace as well. We see that even without the noise added to the traces, TCR is close to 0.93. It means that some tasks, though with very small probability, might still fail to meet

the diversity requirement because the sensing locations are too remote to the majority of participants.

6.6.7 Impact of M

Fig. 6.12e shows the impact of M, the number of sensing tasks, on TTD. As expected, the distance increases with the number of sensing tasks although the increase is limited. Fig. 6.12f shows that TCR decreases with M. Since more tasks are generated, SSP might have to select participants far away to perform the tasks, which leads to a higher assignment decline rate. In addition, since some tasks might be generated in areas with low population of participants, these tasks might fail as well.

6.6.8 Impact of ϵ

We change the value of ϵ in our simulations and evaluate the impact on the performance. Fig. 6.12g shows the results of TTD. We see that with ϵ increasing, TTD decreases. This indicates that a larger ϵ can generate more precise mobility traces that are closer to the original ones. We also observe from Fig. 6.12h that TCR in our scheme is almost identical to that in the baseline approach when ϵ is 4. In addition, we show the average ΔX size in Fig. 6.10. We can see that the number of candidate locations drops from approximately 6.9 to approximately 3.9 with the increase of ϵ. This matches our expectation well. When ϵ is small, the scheme generates larger noise to the participants' mobility traces, hence a larger ΔX. This indicates a better location- privacy protection as well due to more candidate locations. Correspondingly, it will be more difficult for the attacker to infer the participants' true locations.

6.6.9 Impact of δ

δ also has a similar impact on the system performance to ϵ, though we find the system is more sensitive to it. Fig. 6.12i shows the TTD results with different δ values. We can see that there is a relatively big decrease when δ increases from 0.01 to 0.02, and the curve is becoming flat when δ is over 0.02. So this could indicate 0.02 is a good choice of δ for our system. Correspondingly, TCR also generally increases when δ increases but the gain is most observable when we increase δ from 0.01 to 0.02. Recall that a larger δ means a smaller δ-location set and hence worse location privacy. On the other hand, a larger δ can generate mobility traces closer to the original traces, leading to smaller distance errors. We then show the average ΔX size in Fig. 6.11. It can be observed that the ΔX size is very sensitive to the δ value. The size drops dramatically from 15.3 to 4.4 when δ increases from 0.01 to 0.03. To ensure sufficient location privacy, we find that $\delta = 0.02$ can be a good choice for our system.

6.6.10 Impact of α

We vary the value of α, the distance weight ratio between the waiting-time equivalent distance and the true travel distance in our proposed synthetic travel distance model as in Eq. 6.5. Fig. 6.12k and Fig. 6.12l show the results of TTD and TCR, respectively. We can observe that the increase in α only contributes to a small increase in both TTD and TCR. This indicates that our system is consistent with all the distance models that practical systems might have. In addition, the value of α might directly relate to the payment to each participant. A smaller value of α could indicate smaller payments because less weight is assigned to the waiting-time equivalent

distance. Since we do not propose additional payment schemes in our framework, we omit further analysis in this regard.

6.6.11 Impact of β

In some cases, it is strictly required that the desired sensing diversity be achieved to completely avoid any potential transmission interference to PUs. β is such a system parameter that enables the system to adjust the priority. Fig. 6.12m and Fig. 6.12n show the simulation results of TTD and TCR, respectively. Specifically, we see that a larger β leads to an increased TTD value. This is intuitive because a larger β value usually indicates that some sensing tasks are over-fulfilled, i.e., more than the desired number of participants are selected to perform the sensing tasks. The benefit of a larger β is also clearly shown in Fig. 6.12n where we can see that TCR is almost 1 when β is 1.4 and 1 when β is 1.6. In other words, a larger β ensures that all sensing tasks can be fulfilled with enough participants to guarantee sufficient spatial sensing diversity. It is also worth noting that when β is 1, those tasks that are not fulfilled are usually located remotely to most participants. In practice, the SSP can thus dynamically determine the value of β for each sensing task based on the practical demographic properties of the areas where the sensing tasks are located. This strategy ensures that the majority of sensing tasks can be fulfilled and at the same time manages to reduce the unnecessary TTD (or cost) that could be incurred.

6.6.12 Impact of div^*

Lastly, we evaluate the impact of the diversity requirement div^*. Fig. 6.12o shows the results of TTD. Clearly, with a larger div^*, more participants are likely to be selected to fulfill the sensing tasks, thus resulting in a dramatic increase in TTD. Fig. 6.12p shows the change of TCR. We see that a higher diversity order is very

demanding, generating a larger negative impact on TCR in contrast to the results from varying M in Fig. 6.12e. Specifically, the decrease of TCR is found in both the baseline scheme and our scheme.

6.7 Conclusions

Dynamic spectrum access (DSA) has great potential to address worldwide spectrum shortage by enhancing spectrum efficiency. As a key enabler for DSA systems, crowdsourced spectrum sensing (CSS) allows a spectrum sensing provider (SSP) to outsource the sensing of spectrum occupancy to distributed mobile users. In this chapter, we proposed DPSense, a novel framework that allows the SSP to select mobile users for executing spatiotemporal spectrum-sensing tasks without violating the location privacy of mobile users. Detailed evaluations on real location-traces confirmed that DPSense can provide differential location privacy to mobile users while ensuring that the SSP can accomplish spectrum-sensing task assignment with overwhelming probability and also the minimal cost.

Table 6.1: Summary of Notations

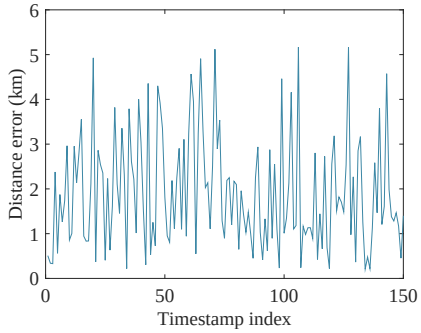
Symbol	**Definition**
N	Total number of participants
M	Total number of sensing tasks
T_j	The jth sensing task
t_j^s	Sensing timestamp for task T_j
R_j	Sensing region for task T_j
div_j^*	Desired sensing diversity order for T_j
$S_{j,k}$	The kth sub-task for T_j
n_j	Number of subtasks for task T_j
L_i	True mobility trace of participant i
ω	Number of timestamps for each mobility trace
$l_{i,\kappa}$	The κth location in L_i
$t_{i,\kappa}$	The κth timestamp in L_i
L_i^o	PIM trace of participant i
L_i^h	Smoothed PIM trace of participant i
$l_{i,\kappa}^o$	The κth location in L_i^o
$l_{i,\kappa}^h$	The κth location in L_i^h
μ	Size of the sliding window
α	Distance weight ratio
β	Diversity order multiplicator

Algorithm 3 Sensing Task Assignment

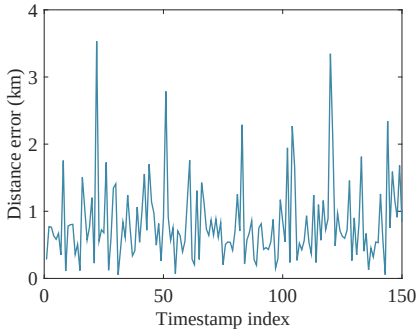
Input: Task set $\mathcal{T}$, subtask sets $\{S_{j,k}\}_{1\le j\le M, 1\le k\le n_j}$, participant set $\mathcal{P}$, PIM trace set $\{L_i^o\}_{i=1}^N$.

Output: $\{b_{j,k}^{i,\kappa}\}_{1\le i\le N, 1\le j\le M, 1\le k\le n_j, \lfloor\mu/2\rfloor+1\le\kappa\le\omega-\lfloor\mu/2\rfloor}$.

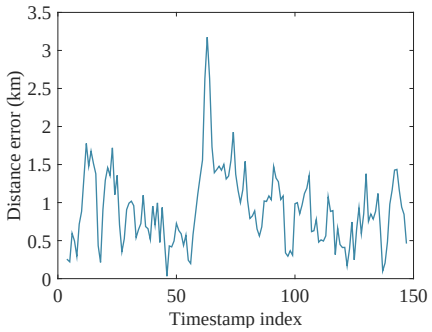
Smooth $\{L_i^o\}_{i=1}^N$ using Algorithm 1 to obtain $\{L_i^h\}_{i=1}^N$.
for all $j \in \mathcal{T}$ **do**
  for all $k \in \{1, \ldots, n_j\}$ **do**
    for all $i \in \mathcal{P}$ **do**
      for all $\kappa \in \{\lfloor\mu/2\rfloor + 1, \ldots, \omega - \lfloor\mu/2\rfloor\}$ **do**
        $b_{j,k}^{i,\kappa} \leftarrow 0$;
        Compute $\mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s)$ as in Eq. (6.5).
      end for
    end for
  end for
  $\mathrm{div_j} \leftarrow \beta \cdot \mathrm{div_j^*}$
  while $\mathrm{div_j} > 0$ **do**
    $\mathsf{dist}^*(l_{i^*,\kappa^*}, l_{j,k^*}^s) = \min\{\mathsf{dist}^*(l_{i,\kappa}, l_{j,k}^s)\}_{\kappa=\lfloor\mu/2\rfloor+1, i\in\mathcal{P}, 1\le k\le n_j}^{\omega-\lfloor\mu/2\rfloor}$;
    $b_{j,k^*}^{i^*,\kappa^*} \leftarrow 1$;.
    Compute $\mathsf{div}_{j,k^*}^{i^*,\kappa^*}$ as in Eq. (6.6).
    $\mathrm{div_j} = \mathrm{div_j} - \mathsf{div}_{j,k^*}^{i^*,\kappa^*}$;
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{i^*\}$;
    $\mathcal{S}_j \leftarrow \mathcal{S}_j \setminus \{S_{j,k^*}\}$;
  end while
end for
return $\{b_{j,k}^{i,\kappa}\}_{1\le i\le N, 1\le j\le M, 1\le k\le n_j, \lfloor\mu/2\rfloor+1\le\kappa\le\omega-\lfloor\mu/2\rfloor}$.



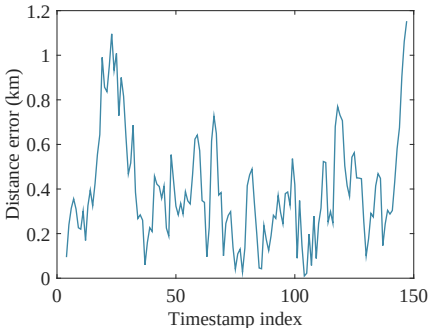
(a) Distance Between the Original Trace and the PIM Trace ($\epsilon = 1$).



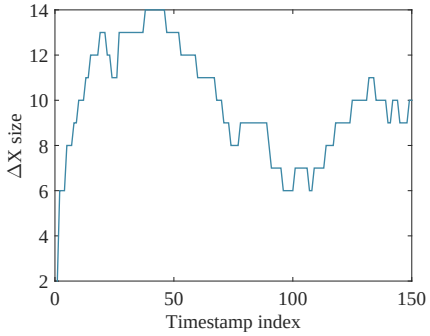
(b) Distance Between the Original Trace and the PIM Trace ($\epsilon = 2$).



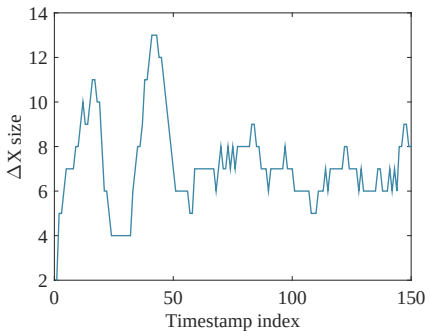
(c) Distance Between the Original Trace and the Smoothed PIM Trace Using the Sliding Window ($\epsilon = 1$).



(d) Distance Between the Original Trace and the Smoothed PIM Trace Using the Sliding Window ($\epsilon = 2$).



(e) ΔX Size ($\epsilon = 1$).



(f) ΔX Size ($\epsilon = 2$).

Figure 6.9: Performance Comparison Using a Single Trace.

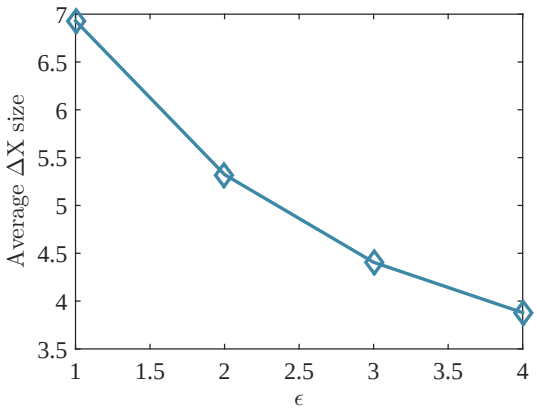


Figure 6.10: ΔX Size for Different ϵ.

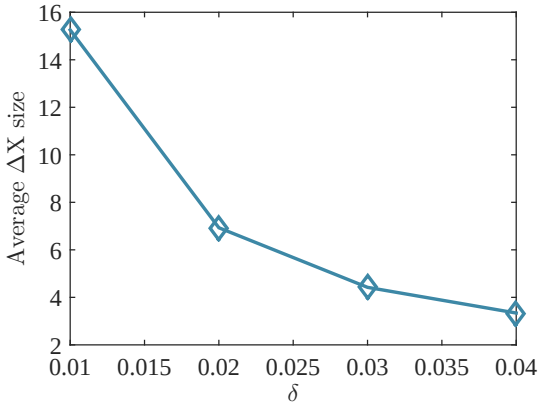
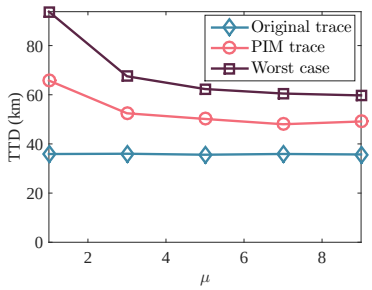
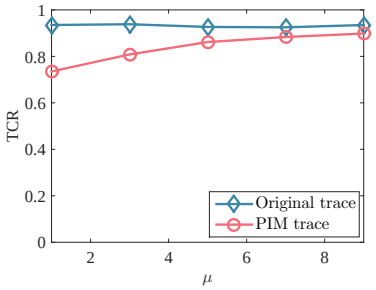


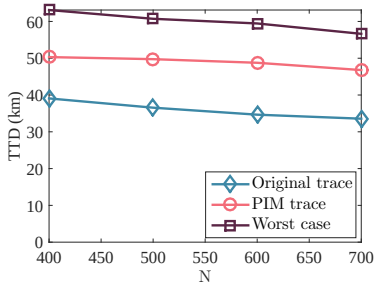
Figure 6.11: ΔX Size for Different δ.



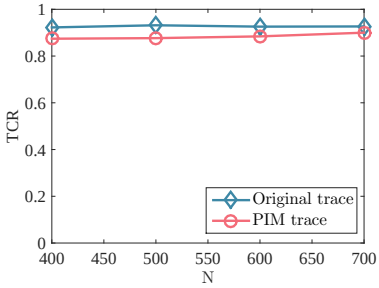
(a) μ vs. TTD.



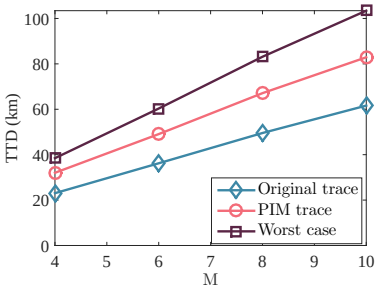
(b) μ vs. TCR.



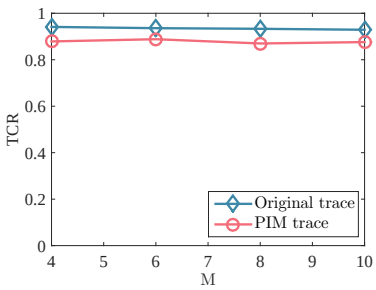
(c) N vs. TTD.



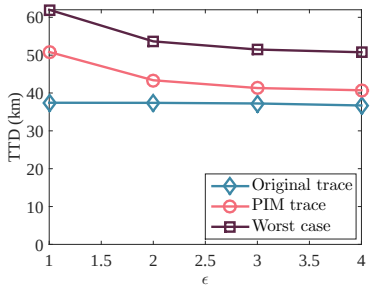
(d) N vs. TCR.



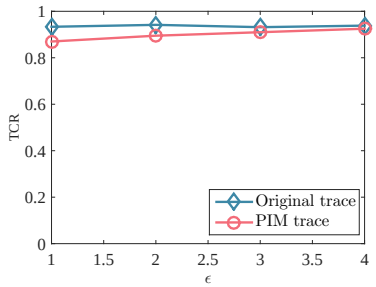
(e) M vs. TTD.



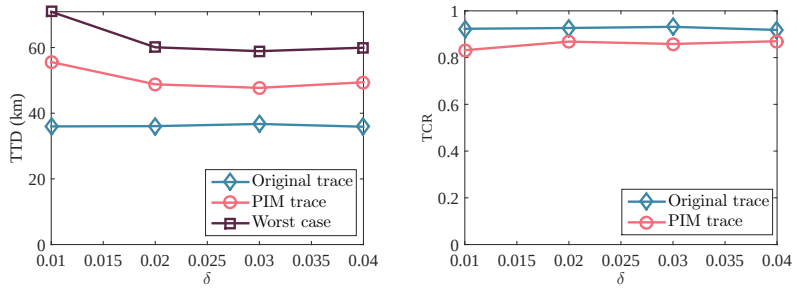
(f) M vs. TCR.



(g) ϵ vs. TTD.

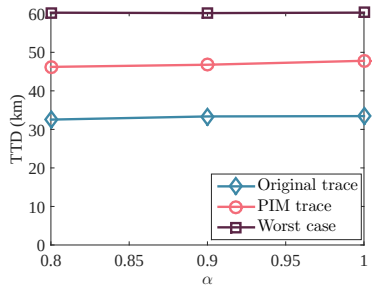


(h) ϵ vs. TCR.

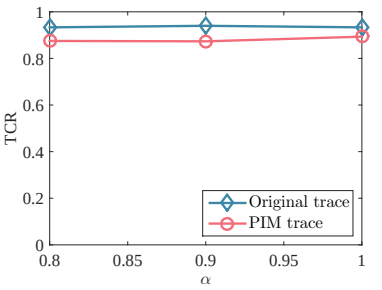


(i) δ vs. TTD.

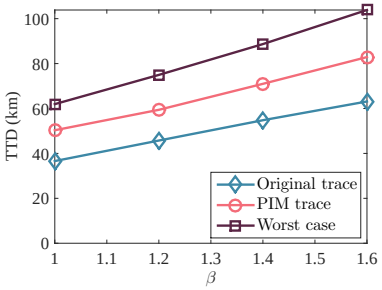
(j) δ vs. TCR.



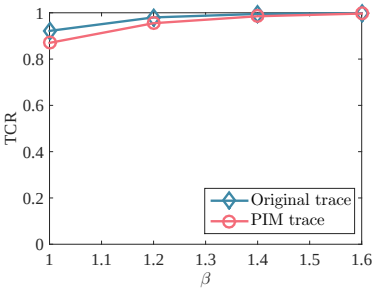
(k) α vs. TTD.



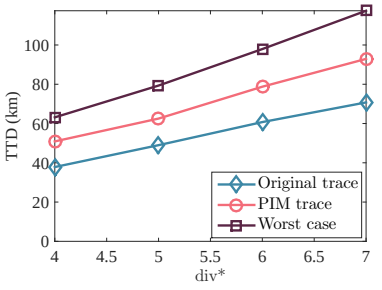
(l) α vs. TCR.



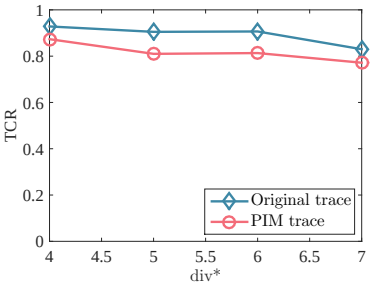
(m) β vs. TTD.



(n) β vs. TCR.



(o) div* vs. TTD.



(p) div* vs. TCR.

Figure 6.12: The Impact of Various Parameters on TTD and TCR.

Chapter 7

CONCLUSION AND FUTURE WORK

DSA is the key paradigm to enable efficient spectrum usage. There are many critical challenges to address before it can be deployed widely in practice. In this dissertation, we focus on one specific security issue: spectrum-misuse detection and one specific privacy issue: location privacy-preserving participant selection in crowdsourcing-based spectrum sensing.

The first part of the dissertation aims to address spectrum-misuse detection. A typical solution is to rely on misuse detectors to sense the spectrum and detect any possible misuse in real time. The transmitter is required to embed certain spectrum permit bits generated by cryptographic tools into his normal transmissions. In this dissertation, we proposed two novel and unique solutions to enable correct, low-intrusive, and fast spectrum-misuse detection. The first solution, *SpecGuard*, embeds spectrum permits spatially in the constellation diagram and the second solution, *SafeDSA*, relies on the temporal gaps (the cyclic prefix) in OFDM symbols to embed the spectrum permits. SpecGuard is simple, easy to implement and SafeDSA is very robust but relies on OFDM. Thorough theoretical analysis and detailed simulations demonstrate that both schemes can achieve correct, low-intrusive and fast detection of spectrum misuse.

The second part of the dissertation aims to provide location-privacy protection for participant selection in crowdsourcing-based spectrum sensing. Since avoiding harmful interference with primary users is the first principle in DSA systems, crowdsourced spectrum sensing is a very promising framework to enable real-time, accurate detection of unused spectrum resources. In this framework, the sensing cost of each

participant is directly associated with the participant's location. Thus, inappropriate handling of this sensitive location information might discourage wide participation. Aiming to protect the location privacy of participants, we proposed two distinct solutions. The first one, *PriCSS*, assumes that the service provider is trusted. We incorporate differential privacy into a reverse auction framework to enable participant selection with location privacy guarantee. The second one, *DPSense*, works even when the service provider is honest but curious. By collecting perturbed mobility traces from participants with a differential location privacy guarantee, the service provider can assign participants according to a probabilistic model. We have conducted thorough theoretical analysis as well as simulations to validate the effectiveness of the two schemes.

Our current effort is still far from perfect. For spectrum-misuse detection, the following directions might be worth exploring. First, it is desired that the spectrum-permit embedding be completely oblivious to the secondary receivers. In our current designs, we require either additional decoding effort or undesired interference to the normal transmissions. If the secondary receiver can simply and successfully decode the normal data transmission without modifications on his PHY-layer hardware, then it is more likely that the scheme can be widely adopted. Second, it is worth studying to see if the idea of SafeDSA can be easily extended to other modulation schemes. For now, SafeDSA relies on the cyclic prefix to embed the spectrum permits. The cyclic prefix, in essence, is a form of timing gap that exits between the symbols in the physical layer. It would be interesting to see if we can find any similar timing-gap notion for other modulations such as the single-carrier modulation and whether this idea can be extended. For the privacy-preserving participant selection, the following directions might be worth exploring. First, we might need to achieve other design objectives other than minimum social cost in practice. For example, for individual

business owners or small enterprises, it is more straightforward to minimize the total payment. For that purpose, the solution could be completely different due to the limitation of theoretical tools available. How to maintain truthfulness is also one key consideration. Second, it would be interesting and challenging to see if we can extend PriCSS to a framework where the service provider is untrusted. There have been many schemes adopting cryptography or third party to conduct privacy-preserving auctions. In light of those works, it is interesting to see if PriCSS can even work without a trusted service provider. Third, it might be interesting and essential to think about alternatives for the probabilistic model we introduced in DPSense. In DPSense, we adopted a linear probabilistic model to characterize the probability of acceptance of a certain task. It is interesting to see if our scheme still delivers reasonable results if we have a different probabilistic model. Additionally, we might also be interested in learning how the system would perform if each participant has a different probabilistic model or simply when the participants' probability cannot be modeled.

REFERENCES

[1] 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.

[2] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.

[3] R. Amici, M. Bonola, L. Bracciale, A. Rabuffi, P. Loreti, and G. Bianchi, "Performance assessment of an epidemic protocol in VANET using real traces," in *CSCW'10*, Savannah, GA, Feb. 2014.

[4] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Retrieved from http://crawdad.org/roma/taxi/20140717, Jul. 2014.

[5] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.

[6] L. Yang, Z. Zhang, B. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *MobiHoc'12*, Hilton Head Island, SC, June 2012.

[7] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "SafeDSA: Safeguard dynamic spectrum access against fake secondary users," in *CCS'15*, Denver, CO, Oct. 2015.

[8] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "SpecGuard: Spectrum misuse detection in dynamic spectrum access systems," in *INFOCOM'15*, Hong-Kong, China, Apr. 2015.

[9] V. Kumar, J. Park, T. Clancy, and K. Bian, "PHY-layer authentication by introducing controlled inter symbol interference," in *CNS'13*, Washington, D.C., Oct. 2013.

[10] V. Kumar, J. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *CCS'14*, Scottsdale, AZ, Nov. 2014.

[11] V. Brik, V. Shrivastava, A. Mishra, and S. Banerjee, "Towards an architecture for efficient spectrum slicing," in *HotMobile'07*, Tucson, AZ, Feb. 2007.

[12] 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.

[13] G. Denker, E. Elenius, R. Senanayake, M. Stehr, and D. Wilkins, "A policy engine for spectrum sharing," in *DySPAN'07*, Dublin, Ireland, Apr. 2007.

[14] S. Liu, L. Greenstein, Y. Chen, and W. Trappe, "ALDO: An anomaly detection framework for dynamic spectrum access networks," in *INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[15] K. Lee, H. Wang, and H. Weatherspoon, "PHY covert channels: Can you see the idles?" in *NSDI'14*, Seattle, WA, Apr. 2014.

[16] X. Wang and D. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *CCS'03*, Washington, D.C., Oct. 2003.

[17] S. Radhakrishnan, A. Uluagac, and R. Beyah, "Realizing an 802.11-based covert timing channel using off-the-shelf wireless cards," in *GLOBECOM'13*, Atlanta, GA, Dec. 2013.

[18] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM'08*, Phoenix, AZ, Apr. 2008.

[19] 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.

[20] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *INFOCOM'13*, Turin, Italy, Apr. 2013.

[21] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *INFOCOM'12*, Orlando, FL, Apr. 2012.

[22] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 106–112, Dec. 2012.

[23] 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*, Turin, Italy, Apr. 2013.

[24] R. Chen, J. Park, and J. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communcations*, vol. 26, no. 1, pp. 25–37, Jan. 2008.

[25] 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.

[26] 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.

[27] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021 white paper," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, Accessed Apr. 28 2017, [Online].

[28] O. Fatemieh, R. Chandra, and C. Gunter, "Secure collaborative sensing for crowdsourcing spectrum data in white space networks," in *DySPAN'10*, Singapore, Singapore, Apr. 2010.

[29] A. Min, X. Zhang, and K. Shin, "Detection of small-scale primary users in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 2, pp. 349–361, Feb. 2011.

[30] Wikipedia, "SHA-1," http://en.wikipedia.org/wiki/SHA-1, Accessed Apr. 18 2016, [Online].

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

[32] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.

[33] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Transactions on Signal Processing*, 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 Transactions on Communications*, vol. 54, no. 5, pp. 841–848, May 2006.

[35] J. Geier, "How to define minimum SNR values for signal coverage," http://www.wi-fiplanet.com/tutorials/article.php/3743986, Accessed Apr. 18 2016, [Online].

[36] Wikipedia, "Orthogonal frequency-division multiplexing," http://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing, Accessed Apr. 18 2016, [Online].

[37] D. Tse and P. Viswanath, *Fundamentals of wireless communications*. Cambridge University Press, 2005.

[38] L. Deneire, B. Gyselinckx, and M. Engels, "Training sequence versus cyclic prefix-a new look on single carrier communication," *IEEE Communications Letters*, vol. 5, no. 7, pp. 292–294, Jul. 2001.

[39] IEEE, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," June 2003.

[40] J. van de Beek, O. Edfors, M. Sandell, S. Wilson, and P. Borjesson, "On channel estimation in OFDM systems," in *VTC'95*, Chicago, IL, Jul. 1995.

[41] J. van de Beek, M. Sandell, and P. Borjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800–1805, Jul. 1997.

[42] M. Sherman, A. Mody, R. Martinez, C. Rodriguez, and R. Reddy, "IEEE standards supporting cogitive radio and networks, dynamic spectrum access, and coexistence," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 72–79, Jul. 2008.

[43] R. van Ne and R. Prasad, *OFDM for wireless multimedia communications*. Artech House, Boston, 2000.

[44] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Transactions on Broadcasting*, vol. 48, no. 3, pp. 223–229, Sept. 2002.

[45] J. Manton, "Optimal training sequences and pilot tones for OFDM systems," *IEEE Communications Letters*, vol. 5, no. 4, pp. 151–153, Apr. 2001.

[46] W. Warner and C. Leung, "OFDM/FM frame synchronization for mobile radio data communication," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 3, pp. 303–313, Aug. 1993.

[47] H. Minn, M. Zeng, and V. Bhargava, "On timing offset estimation for OFDM systems," *IEEE Communications Letters*, vol. 4, no. 7, pp. 242–244, Jul. 2000.

[48] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch, "Geolocation database techniques for incumbent protection in the TV white space," in *DySPAN'08*, Chicago, IL, Oct. 2008.

[49] S. Shellhammer, S. Shankar, R. Tandra, and J. Tomcik, "Performance of power detector sensors of DTV signals in IEEE 802.22 WRANs," in *TAPAS'06*, Boston, MA, Aug. 2006.

[50] T. Zhang, N. Leng, and S. Banerjee, "A vehicle-based measurement framework for enhancing whitespace spectrum databases," in *MobiCom'14*, Maui, HI, Sept. 2014.

[51] A. Nika, Z. Zhang, X. Zhou, B. Zhao, and H. Zheng, "Towards commoditized real-time spectrum monitoring," in *HotWireless'14*, Maui, Hawaii, Sept. 2014.

[52] J. Sun, R. Zhang, X. Jin, and Y. Zhang, "SecureFind: Secure and privacy-preserving object finding via mobile crowdsourcing," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1716–1728, Mar. 2016.

[53] J. Krumm, "A survey of computational location privacy," *Personal Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, Aug. 2009.

[54] R. Shokri, G. Theodorakopoulos, J. Boudec, and J. Hubaux, "Quantifying location privacy," in *S&P'11*, Oakland, CA, May 2011.

[55] W. Wang and Q. Zhang, "Privacy-preserving collaborative spectrum sensing with multiple service providers," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1011–1019, Feb. 2015.

[56] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," in *VLDB'14*, Hangzhou, China, Sept. 2014.

[57] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *ICDCS'16*, Nara, Japan, June 2016.

[58] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 563–571, Aug 2016.

[59] M. Clark and K. Psounis, "Can the privacy of primary networks in shared spectrum be protected?" in *INFOCOM'16*, San Francisco, CA, Apr. 2016.

[60] Y. Dou, K. Zeng, H. Li, Y. Yang, B. Gao, K. Ren, and S. Li, "P^2-SAS: Privacy-preserving centralized dynamic spectrum access system," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 1, pp. 173–187, Jan. 2017.

[61] A. Min, K. Shin, and X. Hu, "Attack-tolerant distributed sensing for dynamic spectrum access networks," in *ICNP'09*, Princeton, NJ, Oct. 2009.

[62] H. Li and Z. Han, "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3554–3565, Nov. 2010.

[63] S. Li, H. Zhu, Z. Gao, X. Guan, and K. Xing, "YouSense: Mitigating entropy selfishness in distributed collaborative spectrum sensing," in *INFOCOM'13*, Turin, Italy, Apr. 2013.

[64] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "ARTSense: Anonymous reputation and trust in participatory sensing," in *INFOCOM'13*, Turin, Italy, Apr. 2013.

[65] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *MDM'14*, Brisbane, Australia, Jul. 2014.

[66] S. He, D. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *INFOCOM'14*, Toronto, Canada, Apr. 2014.

[67] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao, "Reliable diversity-based spatial crowdsourcing by moving workers," in *VLDB'15*, Kohala Coast, HI, June 2015.

[68] D. Deng, C. Shahabi, and L. Zhu, "Task matching and scheduling for multiple workers in spatial crowdsourcing," in *SIGSPATIAL'15*, Seattle, WA, Nov. 2015.

[69] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *MobiCom'12*, Istanbul, Turkey, Aug. 2012.

[70] Z. Feng, Y. Zhu, Q. Zhang, L. Ni, and A. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsoucing," in *INFOCOM'14*, Toronto, Canada, Apr. 2014.

[71] D. Zhao, X. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *INFOCOM'14*, Toronto, Canada, Apr. 2014.

[72] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *INFOCOM'15*, HongKong, China, Apr. 2015.

[73] C. Dwork, "Differential privacy," in *ICALP'06*, Venice, Italy, Jul. 2006.

[74] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *FOCS'07*, Providence, RI, Oct. 2007.

[75] Z. Huang and S. Kannan, "The exponential mechanism for social welfare: Private, truthful, and nearly optimal," in *FOCS'12*, New Brunswick, NJ, Oct. 2012.

[76] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *CCS'15*, Denver, CO, Oct. 2015.

[77] R. Zhu, Z. Li, F. Wu, K. Shin, and G. Chen, "Differentially private spectrum auction with approximate revenue maximization," in *MobiHoc'14*, Philadelphia, PA, Aug. 2014.

[78] R. Zhu and K. Shin, "Differentially private and strategy-proof spectrum auction with approximate revenue maximization," in *INFOCOM'15*, HongKong, China, Apr. 2015.

[79] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *INFOCOM'16*, San Francisco, CA, Apr. 2016.

[80] Y. Selen, H. Tullberg, and J. Kronander, "Sensor selection for cooperative spectrum sensing," in *DySPAN'08*, Chicago, IL, Oct. 2008.

[81] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, *Algorithmic game theory*. Cambridge University Press, 2007.

[82] V. Vazirani, *Approximation algorithms*. Springer, 2012.

[83] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *MobiHoc'09*, New Orleans, LA, May 2009.

[84] A. Archer and E. Tardos, "Truthful mechanisms for one-parameter agents," in *FOCS'01*, Las Vegas, NV, Oct. 2001.

[85] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT'06*, St. Petersburg, Russia, May 2006.

[86] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar, "Differentially private combinatorial optimization," in *SODA'10*, Austin, TX, Jan. 2010.

[87] X. Jin, R. Zhang, Y. Chen, T. Li, and Y. Zhang, "DPSense: Differentially private crowdsourced spectrum sensing," in *CCS'16*, Vienna, Austria, Oct. 2016.

[88] D. Duan, L. Yang, and J. Principe, "Cooperative diversity of spectrum sensing for cognitive radio systems," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3218–3227, June 2010.

[89] B. Hecht and D. Gergle, "On the "localness" of user-generated content," in *CSCW '10*, Savannah, GA, Feb. 2010.

[90] Wikipedia, "Google now," https://en.wikipedia.org/wiki/Google_Now, Accessed Apr. 18 2017, [Online].

[91] R. Gandhi, S. Khuller, and A. Srinivasan, "Approximation algorithms for partial covering problems," *J. Algorithms*, vol. 53, no. 1, pp. 55–84, Oct. 2004.