SafeDSA: Safeguard Dynamic Spectrum Access against Fake Secondary Users

Xiaocong Jin
Arizona State University
xcjin@asu.edu

Jingchao Sun
Arizona State University
jcsun@asu.edu

Rui Zhang
University of Hawaii
ruizhang@hawaii.edu

Yanchao Zhang
Arizona State University
yczhang@asu.edu

ABSTRACT

Dynamic spectrum access (DSA) is the key to solving worldwide wireless spectrum shortage. In a DSA system, unlicensed secondary users can opportunistically use a spectrum band when it is not used by the licensed primary user. The open nature of the wireless medium means that any secondary user can freely use any given spectrum band. Secondary-user authentication is thus essential to ensure the proper operations of DSA systems. We propose SafeDSA, a novel PHY-based scheme for authenticating secondary users in DSA systems. In SafeDSA, the secondary user embeds his spectrum-use authorization into the cyclic prefix of each physical-layer symbol, which can be detected and authenticated by a verifier. In contrast to previous work, SafeDSA achieves robust and efficient authentication of secondary users with negligible impact on normal data transmissions. We validate the efficacy and efficiency of SafeDSA through detailed MATLAB simulations and USRP experiments. Our results show that SafeDSA can detect fake secondary users with a maximum false-positive rate of 0.091 and a negligible false-negative rate based on USRP experiments.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: Security and Protection

Keywords

Dynamic spectrum access; secondary user authentication; OFDM; cyclic prefix

1. INTRODUCTION

Dynamic spectrum access (DSA) is the key to solving worldwide wireless spectrum shortage. According to the Cisco's Virtual Networking Index [1], global mobile devices in 2014 have surpassed 7.4 billion, and global mobile data traffic will increase nearly tenfold between 2014 and 2019. On the one hand, the industrial, scientific and medical (ISM) spectrum bands and cellular spectrum bands are becoming too congested to accommodate the explosive mobile data traffic. On the other hand, large portions of the licensed wireless spectrum are highly underutilized. DSA is an emerging spectrum-sharing paradigm. In a DSA system, unlicensed secondary users can opportunistically use a spectrum band when it is not used by the licensed primary user.

Secondary-user authentication is essential in DSA systems. The open nature of the wireless medium means that any secondary user can freely use any given spectrum band. How can we authenticate that a secondary user has been authorized to use a given spectrum band in the given area and time duration? Without such secondary-user authentication, legitimate secondary users who often have to pay for spectrum access may suffer severe interference from illegitimate secondary users and thus be discouraged from further using DSA systems. In addition, licensed primary users may have no economic incentives to participate in DSA systems if there are insufficient secondary users paying for spectrum access. The envisioned bright future of DSA systems can thus be severely jeopardized.

There have been recent efforts [2–5] 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 [2] or mobile crowdsourcing users [5]. 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 [2] generates physical-layer cyclostationary features; P-DSA [3] adds controlled inter-symbol interference; FEAT [4] intentionally tunes the frequency offset; SpecGuard [5] explores dynamic power control. Although these schemes can all detect fake secondary users with very low false positives and negatives, Gelato and FEAT have high computational overhead, and SpecGuard is highly dependent on the FCC power constraint imposed on any spectrum band.

We propose SafeDSA, 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 communications systems to eliminate the inter-symbol interference

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CCS'15, October 12–16, 2015, Denver, Colorado, USA.
© 2015 ACM. ISBN 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813717.

from previous symbols and simplify frequency-domain processing in multipath environments [6]. 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 [4]. SafeDSA is thus very feasible for both dedicated, resourceful verifiers [2] and resource-constrained mobile crowd-verifiers [5]. 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 paper is organized as follows. Section 2 briefs the related work. Section 3 introduces the system and adversary models. Section 4 outlines the background on the cyclic prefix and OFDM underlying SafeDSA. Section 5 details the SafeDSA design. Section 6 analyzes the theoretical performance of SafeDSA. Section 7 evaluates SafeDSA through detailed MATLAB simulations. Section 8 reports the performance of SafeDSA through USRP experiments. Section 9 concludes this paper.

2. RELATED WORK

This section reviews the prior work most related to SafeDSA.

The work in [7–9] proposes to equip secondary users with tamper-resistant wireless transceivers to enforce spectrum policies and prevent them from illegitimately using the spectrum. Such tamper-resistant devices are expensive to build and subject to capable attacks. In contrast, SafeDSA does not require any tamper-resistant wireless transceiver on secondary users.

The work in [10] uses a dedicated sensor network to perform spatially distributed power measurements for detecting illegitimate secondary users. SafeDSA avoids deploying and maintaining such a distributed sensor network.

There has been some work [11–13] to construct a physical-layer covert channel which is not easily detectable by the adversary. Although SafeDSA also embeds information into physical-layer signals, it does 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., [14–16]) 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 SafeDSA. Authors in [17, 18] 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 [19] 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 paper. Another line of work [20–22] 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, SafeDSA aims at authenticating secondary users who may or may not be authorized to use the channel.

As said, the schemes in [2–5] are all PHY-based approaches for authenticating secondary users and most germane to SafeDSA. As the seminal work, Gelato [2] targets OFDM, the prevailing technology for wireless communications. In Gelato, every secondary user embeds a spectrum permit by intentionally creating cyclostationary features [23] 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 [24]. P-DSA [3] 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 [4] 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. Finally, SpecGuard [5] explores dynamic power control at the secondary user to contain the spectrum permit in physical-layer signals. There are three techniques in SpecGuard [5]. The performance of the first two techniques are highly dependent on the FCC-specified maximum transmission power on each spectrum band, and the third technique requires the secondary sender to completely trust the secondary receiver for sharing the spectrum permit. SafeDSA neither reduces the data throughput nor has the drawbacks of SpecGuard. It is also computationally much more efficient than FEAT.

3. SYSTEM AND ADVERSARY MODELS

3.1 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.

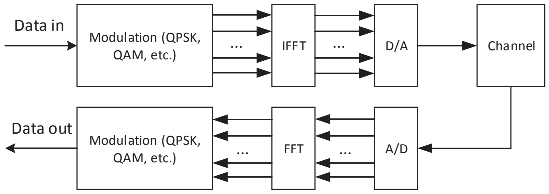


Figure 1: A typical OFDM framework.

- *Verifier*: A SafeDSA verifier is not engaged in data communications with the secondary user. Instead, it passively eavesdrops on the secondary user's transmissions 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 [2] or resource-constrained mobile crowdsourcing workers (referred to as mobile crowd-verifiers) as in [5].

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 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 transmissions, 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.

4. 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 subcarrier 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 [25].

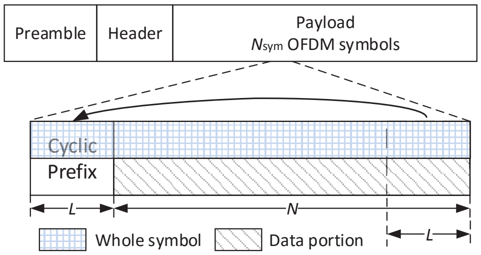


Figure 2: A general OFDM frame structure.

A cyclic prefix refers to prefixing a symbol with a repetition of the end. The concept traditionally roots in orthogonal frequency-division multiplexing (OFDM) [26], and now has its wide applications in single carrier systems [27] as well to improve the resilience to multipath effect. Consider the typical OFDM framework in Fig. 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 structure with N_{sym} OFDM data symbols is shown in Fig. 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 [28]. 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 [29, 30].

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 (guide 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 [31], in which the cyclic prefix length can be set to 1/4, 1/8, 1/16 and 1/32 times the OFDM symbol length.

5. SAFEDSA DESIGN

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

5.1 Spectrum Permit Construction

The spectrum permits in SafeDSA are similar to those in [2,4,5]. 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

Table 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)
α	channel delay spread (measured in samples)
m	expansion ratio of the cyclic prefix length
n	compression ratio of the cyclic prefix length

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 $\gamma \geq 1$ 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 [32]. The operator selects a random number n_γ and recursively computes $n_i = h(n_{i+1} \parallel \text{addr}_{\text{SU}}), \forall i \in [0, \gamma - 1]$, where addr_{SU} denotes the hardware address of the secondary user. Next, the operator sends n_γ securely to the secondary user through traditional security mechanisms such as TLS [33]. Finally, the secondary user recursively computes $\{n_1, \dots, n_{\gamma-1}\}$ in the same way and uses $n_i, \forall i \in [1, \gamma]$, 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 $i \in [1, \gamma]$ 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 γ 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 5.3.

5.2 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 [34]. 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. 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. 2 for the ease of scheme description. We assume that the multipath channel has a delay spread of α (measured in samples) and summarize the key design parameters in Table 1.

5.2.1 Permit Encoding

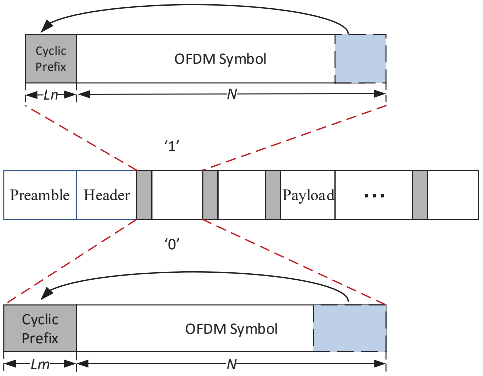


Figure 3: Mapping of permit bits to cyclic prefix length ($M = 2$).

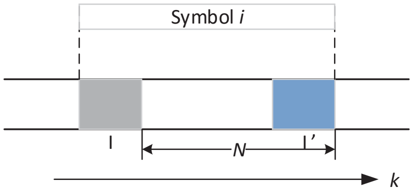


Figure 4: Data dependency evaluation of OFDM symbols.

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 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 α/L. The mapping of the permit bits is illustrated in Fig. 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 paper, 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. Hence, to avoid decreasing the data throughput, we require $p_0 m + p_1 n \leq 1$. Hence, by assuming that p_0 and p_1 are both 0.5, we have the following constraint set for parameter configurations:

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

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.

5.2.2 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 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. 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 [29]:

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

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. (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:

$$D = \sum_{p=0}^{N_{\text{sym}}-1} \sum_{k=1}^{nL} |r((N+L')p+k) - r((N+L')p+k +N)|,\ L' \in \{mL, nL\}. \quad (3)$$

The second metric is the correlation C, defined as:

$$C = \sum_{p=0}^{N_{\text{sym}}-1} \sum_{k=1}^{nL} r((N+L')p+k)r^*((N+L')p+k +N),\ L' \in \{mL, nL\}. \quad (4)$$

According to Eq. (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 pairwise 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 could be used (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 [35]. Specifically, we first define

$$P(p) = \sum_{k=1}^{L'} r((N+L')p+k)r^*((N+L')p+k +N),\ L' \in \{mL, nL\},\ p \in [0, N_{\text{sym}}-1] \quad (5)$$

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

$$R(p) = \sum_{k=1}^{L'} |r((N+L')p+k)|^2,\ L' \in \{mL, nL\},\ p \in [0, N_{\text{sym}}-1]. \quad (6)$$

The metric T is then defined as

$$T = \frac{|\sum_{p=0}^{N_{\text{sym}}-1} P(p)|^2}{(\sum_{p=0}^{N_{\text{sym}}-1} R(p))^2}, \quad (7)$$

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 7.

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

$$\hat{L} = \text{argmax}_{L'}|C| \text{ or } \hat{L} = \text{argmin}_{L'}D \text{ or } \hat{L} = \text{argmax}_{L'}T\ , \quad (8)$$

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) to (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 8. 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 7 and Section 8 with MATLAB simulations and USRP experiments, respectively.

5.2.3 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. (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. 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.

5.3 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

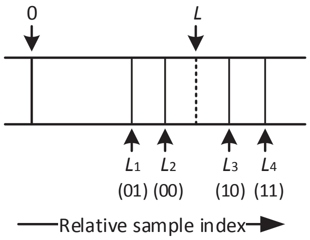


Figure 5: Extension of M with Gray coding.

a one-way hash chain is used to construct the spectrum permits, the operator additionally sends the hash value n_0 (see Section 5.1) 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 5.2.2. 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.

6. ANALYSIS

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

6.1 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 [4], 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.

6.2 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 [36–38]. 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 [30] 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.

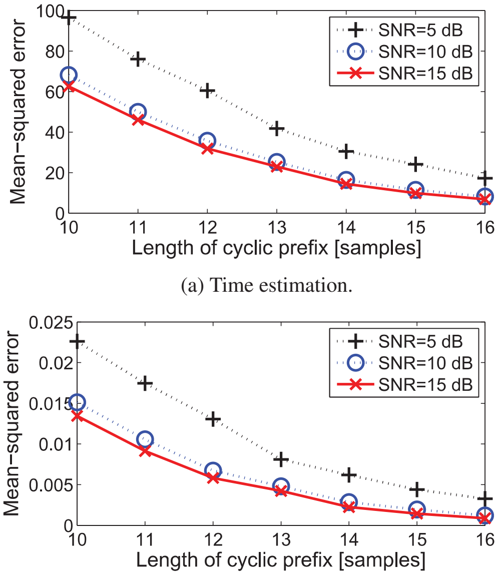
6.3 Impact on Frequency/Timing Offset Estimation

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 [39] relies on pilot symbols known to the receiver to perform the estimations. Similar as the channel estimation, the shortened cyclic prefix length will not impact the estimation performance. Other work such as [29] 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 [29] 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. 6 in the form of 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 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 [29] 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.

6.4 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.



(a) Time estimation.

(b) Frequency estimation.

Figure 6: Performance of the time and frequency estimators for the AWGN channel.

6.4.1 *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.

6.4.2 *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.

7. 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:

$$d = L - nL = mL - L. \quad (9)$$

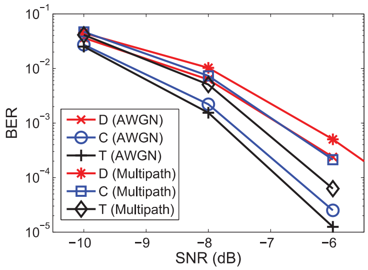


Figure 7: Comparing data dependency metrics (d=1).

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 [4] and SpecGuard [5]. We do not choose Gelato [2] 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 $N_{\mathrm{sym}} = 25$. 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.

7.1 Data Dependency Metric

Recall that in Eq. (3), Eq. (4) and Eq. (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 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. 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.

7.2 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. 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), Eq. (4), and Eq. (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. (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. 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. 8, which proves its correctness.

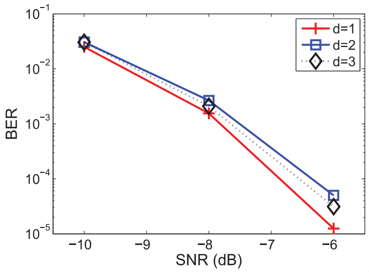


Figure 8: Permit BER for different ds.

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 but not 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. 8 shows that the three curves are close to each other with a low BER overall. For the rest of the paper, unless otherwise mentioned, d is set to 1.

7.3 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. 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. 10. For ease of evaluation, we simply let $\lfloor w/\beta \rfloor$ in Section 5.3 be 1, i.e., only one copy of candidate 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. 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 is 13, the false-positive rate is generally much worse than that when N is 25. However, since usually wireless communications are conducted when SNR is above 0 dB [40], SafeDSA can achieve desirable detection performances in this regard. On the other hand, false-negatives occur

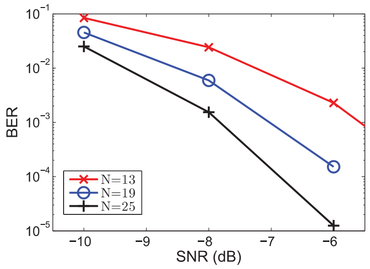


Figure 9: Permit BER for different frame lengths.

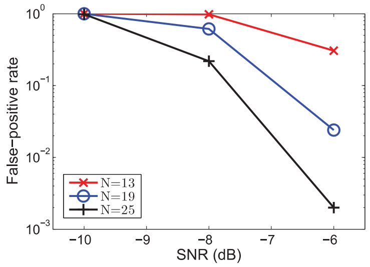


Figure 10: False-positive rate for different frame lengths.

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.

7.4 The Value of M

We also evaluate permit-decoding performance for different values of M in Fig. 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 such low.

7.5 Comparison with Related Work

In this section, we compare SafeDSA with FEAT [4] and SpecGuard [5]. 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 [5], 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 [5], 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. 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 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.

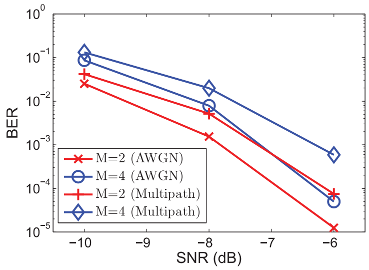


Figure 11: Permit BER for different Ms.

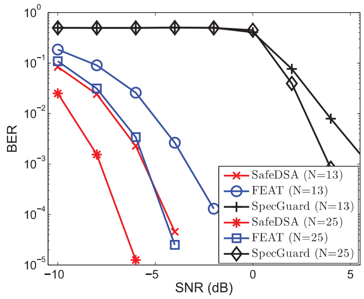


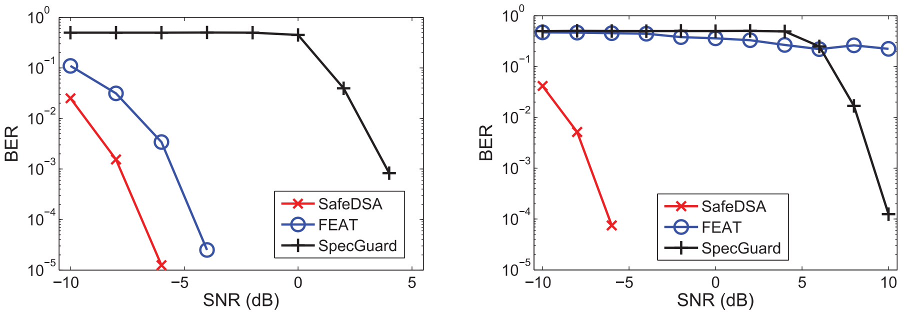
Figure 13: Permit BER comparison for different frame lengths.

We then compare the three schemes for different frame lengths in the AWGN channel. Fig. 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.

Lastly, we compare the three schemes' impact on normal data transmissions. 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 transmissions. 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 transmissions. 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. 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.

8. 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



(a) AWGN channel. (b) Multipath Rayleigh fading channel.

Figure 12: Permit BER comparison for different channels.

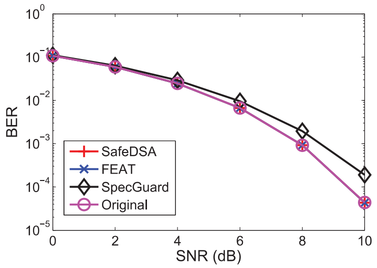
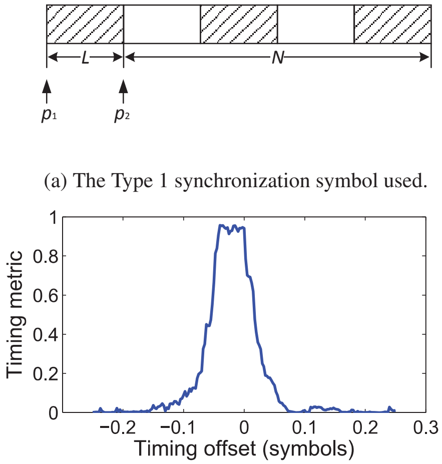


Figure 14: Data BER comparison in AWGN channel.

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 7.

Different from the 802.11 standard introduced earlier, our USRP N210 transceiver only uses two preambles as discussed in [35] 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 paper at the receiver end. As discussed in Section 5, 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 [35]. 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.

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 [35]. Fig. 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. 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. (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.



(a) The Type 1 synchronization symbol used.

(b) The plateau effect (SNR= 20 dB).

Figure 15: The plateau effect when using the timing offset estimation method in [35].

Fig. 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

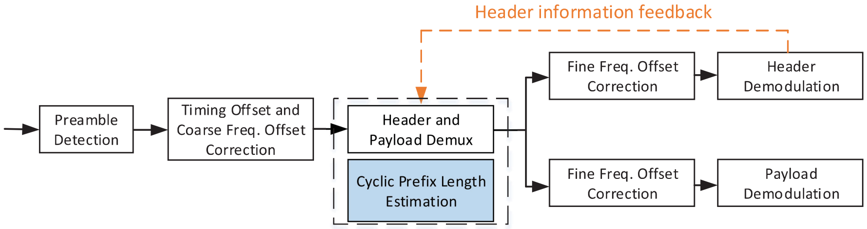

Header information feedback
Preamble Detection
Timing Offset and Coarse Freq. Offset Correction
Header and Payload Demux
Cyclic Prefix Length Estimation
Fine Freq. Offset Correction
Header Demodulation
Fine Freq. Offset Correction
Payload Demodulation

Figure 16: The flowchart of the SafeDSA receiver design in GNU Radio.

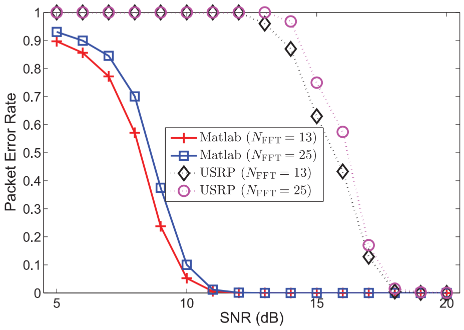


Figure 17: Packet error rate comparison using USRP benchmark transceivers and MATLAB simulations.

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.

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. 17. We vary N 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∼20 dB) match with the range in Fig. 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 from 13 to 25 and varying M from 2 to 4. As expected, when N is 25, the permit BER is always 0 or too low to observe. When N 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. 9 and Fig. 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 [41] to ensure consistently small sample offsets. These investigations are left as future work.

9. CONCLUSION

This paper 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.

10. ACKNOWLEDGMENTS

This work was partially supported by the US National Science Foundation under grants CNS-1320906, CNS-1421999, and CNS-1117462.

11. REFERENCES

[1] "Cisco Visual Networking Index: Global mobile data traffic forecast update 2014-2019 white paper," `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html`, [Online].

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

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

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

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

[6] A. Goldsmith, "Wireless communications," pp. 172-197, 2005.

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

[8] 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.

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

[10] 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.

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

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

[13] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "Realizing an 802.11-based Covert Timing Channel Using Off-The-Shelf Wireless Cards," in *GLOBECOM'13*, Altanta, GA, Dec. 2013.

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

[15] 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.

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

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

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

[19] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: attacks and countermeasures," *INFOCOM'13*, Turin, Italy, Apr. 2013.

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

[21] 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.

[22] 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.

[23] J. Sun, D. Qu, T. Jiang, G. Zhong, and J. Guo, "Low overhead cyclostationary signatures based on hopping subcarrier in OFDM-based dynamic spectrum access networks," *VTC Fall'11*, San Francisco, CA, Sept. 2011.

[24] 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.

[25] "Orthogonal frequency-division multiplexing," `http://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing`, [Online].

[26] D. Tse, and P. Viswanath, "Fundamentals of wireless communications," 2005.

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

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

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

[30] J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Börjesson, "On channel estimation in OFDM systems," *Vehicular Technology Conference, 1995 IEEE 45th*, vol. 2, pp. 815-819, Jul. 1995.

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

[32] "SHA-1," `http://en.wikipedia.org/wiki/SHA-1`, [Online].

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

[34] R. van Nee, and R. Prasad, "OFDM for wireless multimedia communications," Artech House, Boston, pp. 46, 2000.

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

[36] C. Sinem, E. Mustafa, P. Anuj, and B. Ahmad, "Channel estimation techniques based on pilot arrangement in OFDM systems," *Broadcasting, IEEE Transactions on*, vol. 48, no. 3, pp. 223-229, Sept. 2002

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

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

[39] W. D. Warner, and C. Leung, "OFDM/FM frame synchronization for mobile radio data communication," *Vehicular Technology, IEEE Transactions on*, vol. 42, no. 3, pp. 302-313, Aug. 1993.

[40] "Design considerations," `http://www.cisco.com/c/en/us/td/docs/wireless/technology/mesh/7-3/design/guide/Mesh/Mesh_chapter_011.pdf`, [Online].

[41] H. Minn, M. Zeng, and V. K. Bhargava, "On timing offset estimation for OFDM systems," *Communication Letters, IEEE*, vol. 4, no. 7, pp. 242-244, July 2000.