Authenticating Primary Users' Signals in Cognitive Radio Networks via Integrated Cryptographic and Wireless Link Signatures

Yao Liu, Peng Ning
Department of Computer Science
North Carolina State University
Raleigh, NC
{*yliu20, pning*}*@ncsu.edu*

Huaiyu Dai
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC
hdai@ncsu.edu

***Abstract*—To address the increasing demand for wireless bandwidth, cognitive radio networks (CRNs) have been proposed to increase the efficiency of channel utilization; they enable the sharing of channels among secondary (unlicensed) and primary (licensed) users on a non-interference basis. A secondary user in a CRN should constantly monitor for the presence of a primary user's signal to avoid interfering with the primary user. However, to gain unfair share of radio channels, an attacker (e.g., a selfish secondary user) may mimic a primary user's signal to evict other secondary users. Therefore, a secure primary user detection method that can distinguish a primary user's signal from an attacker's signal is needed. A unique challenge in addressing this problem is that Federal Communications Commission (FCC) prohibits any modification to primary users. Consequently, existing cryptographic techniques cannot be used directly.**

In this paper, we develop a novel approach for authenticating primary users' signals in CRNs, which conforms to FCC's requirement. Our approach integrates cryptographic signatures and wireless link signatures (derived from physical radio channel characteristics) to enable primary user detection in the presence of attackers. Essential to our approach is a *helper node* placed physically close to a primary user. The helper node serves as a "bridge" to enable a secondary user to verify cryptographic signatures carried by the helper node's signals and then obtain the helper node's authentic link signatures to verify the primary user's signals. A key contribution in our paper is a novel physical layer authentication technique that enables the helper node to authenticate signals from its associated primary user. Unlike previous techniques for link signatures, our approach explores the geographical proximity of the helper node to the primary user, and thus does not require any training process.

***Keywords*-cognitive radio networks; primary user detection; link signatures.**

I. Introduction

The proliferation of emerging wireless applications requires a better utilization of radio channels [4]. To address the increasing demand for wireless bandwidth, cognitive radio networks (CRNs) have been proposed to increase the efficiency of channel utilization under the current static channel allocation policy [17]. They enable unlicensed users to use licensed channels on a non-interference basis, thus serve as a solution to the current low usage of radio channels [8]. For example, IEEE 802.22 Standard on Wireless Regional Area Networks (WRANs) employs cognitive radio to allow the sharing of geographically unused channels allocated to television broadcast services, and therefore bring broadband access to hard-to-reach low-population-density areas (e.g., rural environments) [9].

In CRNs, there are two types of users: *primary users* and *secondary users* [17]. Primary users are licensed users who are assigned with certain channels, and secondary users are unlicensed users who are allowed to use the channels assigned to a primary user only when they do not cause any harmful interference to the primary user [17]. For example, in IEEE 802.22 WRANs, TV transmission towers are primary users, and radio devices that use TV channels for communication are secondary users.

An essential issue in CRNs is *primary user detection*, in which a secondary user monitors for the presence of a primary user's signal on target channels [4]. If a primary user's signal is detected, the secondary user should not use those channels to avoid interfering with the transmission of the primary user.

Existing methods for primary user detection can be categorized as *energy detection* and *feature detection* [17]. In energy detection methods (e.g., [30]), any captured signal whose energy exceeds a threshold is identified as a primary user's signal. In feature detection methods (e.g., [12], [25], [26], [29], [37]), secondary users attempt to find a specific feature of a captured signal, such as a pilot, a synchronization word, and cyclostationarity. If a feature is detected, then the captured signal is identified as a primary user's signal.

Due to the open nature of wireless communications and the increasingly available software defined radio platforms (e.g., Universal Software Radio Peripherals (USRPs) [10]), it is necessary to consider potential threats to normal operations of CRNs. Indeed, CRNs do face several threats. In particular, an attacker may transmit with high power or mimic specific features of a primary user's signal (e.g., use the same pilots or synchronization words) to bypass the existing primary user detection methods [4]. Consequently, secondary users may incorrectly identify the attacker's signal as a primary user's signal and do not use relevant channels. Such attacks are called *primary user emulation (PUE) attacks* [4].

It is necessary to have a secure primary user detection method that can identify a primary user's signal in the presence of attackers. At first glance, a cryptographic signature seems to be a good candidate for this task. Unfortunately, CRNs face a unique constraint that prevents it from being employed. Specifically, Federal Communications Commission (FCC) states that "no modification to the incumbent system (i.e., primary user) should be required to accommodate opportunistic use of the spectrum by secondary users" [6]. As a result, any solution that requires changes to primary users, such as enhancing primary users' signals with cryptographic signatures, is not desirable.

There has been a recent attempt that uses a location distinction approach to distinguish between a primary user's signal and an attacker's signal [4]. Specifically, this approach uses received signal strength (RSS) measurements to estimate the location of the source of a signal, and then determines if the signal is from the (static) primary user based on the known location of the primary user [4]. However, as indicated in [23], RSS based location distinction can be easily disrupted if an attacker uses array antennas to send different signal strengths in different directions simultaneously. Moreover, it requires multi-node collaboration, which is expensive in terms of bandwidth and energy.

Link signatures (i.e., radio channel characteristics such as channel impulse responses) have been developed recently to obtain more secure and robust location distinction [23], [38]. Unfortunately, it remains non-trivial to exploit link signature based location distinction approach for primary user detection in the presence of attackers. In particular, a receiver needs to know a transmitter's historical link signatures in order to verify if a newly received signal is from the transmitter. In CRNs, however, it is impossible for a secondary user to know a primary user's historical link signatures, unless the secondary user can first authenticate whether a signal is from the primary user or not.

In this paper, we develop a novel approach that integrates traditional cryptographic signatures and link signatures to enable primary user detection in the presence of attackers. Our approach does not require any change to primary users, and thus follows the FCC constraint properly.

A key component of our approach is a *helper node* placed close (and physically bound) to a primary user. Though we cannot modify any primary user due to the FCC constraint, we can put necessary mechanisms on each helper node, including the use of cryptographic signatures. Moreover, since the helper node is placed very close to the primary user, their link signatures observed by a secondary user are very similar to each other. The helper node thus serves as a "bridge" that enables a secondary user to first verify the cryptographic signatures included in the helper node's signals, then learn the helper node's authentic link signatures, and finally verify the primary user's link signatures. In other words, our approach properly integrates cryptographic signatures and wireless link signatures to enable primary user detection in CRNs in the presence of attackers.

The contributions of this paper are summarized below:

- We develop a new primary user detection method that integrates cryptographic signatures with wireless link signatures to distinguish a primary user's signal from an attacker's signal. Our method conforms to the FCC's requirement of not modifying primary users. Unlike the previous approach [4], our method does not require the deployment of a monitoring network, and thus avoids the weakness of the previous approach.
- We develop a novel physical-layer authentication technique that enables a helper node to authenticate signals from its associated primary user. Unlike previous proposals for link signatures, our approach explores the geographical proximity of the helper node to the primary user rather than historical link signatures. A key consequence is that our method does not require any training process.
- We evaluate the effectiveness of our method through both theoretical analysis and experiments using real-world link signatures obtained from the CRAWDAD data set [22]. Moreover, we demonstrate the feasibility of our proposed method by a prototype implementation on a software-defined radio platform [10].

The rest of the paper is organized as follows. Section II gives background information about link signatures. Section III explains our assumptions and threat model. Section IV gives an overview of our method. Sections VI and V present the primary user detection at a helper node and a secondary user, respectively. Section VII discusses the experimental evaluation. Section VIII describes a prototype implementation of our method. Section IX discusses related work, and Section X concludes this paper.

II. Preliminaries

In this section, we provide some preliminary information on link signatures, which will be used for primary user detection.

Radio signal generally propagates in the air over multiple paths due to reflection, diffraction, and scattering [23]. Therefore, a receiver usually receives multiple copies of the transmitted signal (See Figure 1). Since different paths have different distances and path losses, signal copies travel on multiple paths typically arrive at the receiver at different times and with different attenuations [23]. The sum of those signal copies forms the received signal. For the sake of presentation, we refer to a signal copy that travels along one path as a *multipath component*. For example, in Figure 1, signals s_1, s_2, s_3, and s_4 are multipath components.

Multipath effect might be reduced by using directional antennas. However, directional antennas usually cannot provide perfect laser-like radio signals. For example, the beamwidth of a 3-element Yagi Antenna, the most common type of directional antennas, is 90 degrees in the vertical plane and 54 degrees in the horizontal plane [18]. Thus, it is in general hard to completely eliminate multipath effect. For long distance transmission, the amount of multipath effect seen by a receiver may be much more due to the reduced focusing power at the receiver [2].

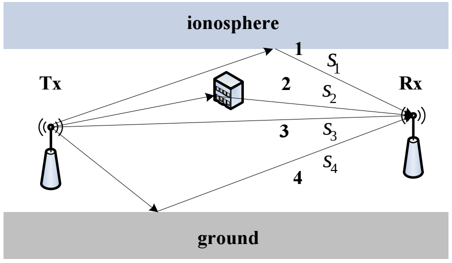
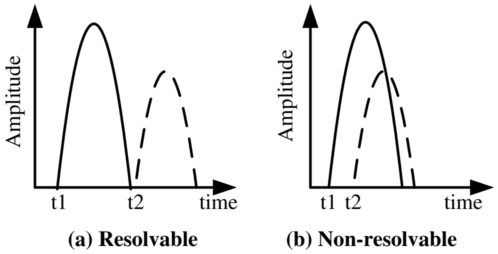


Figure 1. Example of a multipath effect. The wireless signal sent by transmitter Tx is reflected by the ionosphere, a building, and the ground. Thus, radio waves propagate over paths 1, 2, 3, and 4. The receiver Rx receives signal copies s_1, s_2, s_3, and s_4 from paths 1, 2, 3, and 4, and the received signal is the sum of all signal copies.

Note that a multipath component herein refers to a resolvable multipath component (i.e., the arrival of a multipath component does not interfere with that of its subsequent multipath component). Figure 2 is an example that shows the difference between resolvable and non-resolvable multipath components.



(a) Resolvable **(b) Non-resolvable**

Figure 2. Resolvable and non-resolvable multipath components. In (a), the arrivals of two multipath components do not interfere with each other. Therefore, they are resolvable. In (b), the arrival of the second multipath component interferes with that of the first multipath component. Therefore, they are non-resolvable.

A radio channel consists of multiple paths from a transmitter to a receiver, and each path of the channel has a response (e.g., distortion and attenuation) to the multipath component traveling on it [23]. For convenience, we call the response to each multipath component a *component response*. Essentially, the *channel impulse response* is formed by the superposition of many component responses, each representing a single path [23]. Therefore, the channel impulse response, denoted by $h(\tau)$, is given by

$$h(\tau) = \sum_{l=1}^{L} a_l e^{j\phi_l} \delta(\tau - \tau_l), \tag{1}$$

where L is the total number of multipaths, $\delta(\tau)$ is the Dirac delta function, and a_l, ϕ_l, and τ_l are the channel gain, the phase, and the time delay of the l-th multipath component, respectively [23].

If a transmitter moves from one place to another, the multiple paths from the transmitter to the receiver change, and thus the channel impulse response also changes. As a result, the channel impulse responses can be used to determine whether the transmitter changes its location or not. A channel impulse response is referred to as a *link signature* [23]. A location distinction algorithm using link signatures has been proposed in [23]. Specifically, a history of n link signatures are measured and stored while the transmitter is not moving. For a newly measured link signature, the receiver computes the distance between the newly measured link signature and the historical link signatures. If the distance is larger than a threshold, then a location change is detected.

III. Assumptions and Threat Model

Our system consists of primary users and secondary users. A primary user is assumed to be at a fixed location (e.g., a TV broadcast tower) [4]. As stated by FCC, TV stations and radio infrastructures should maintain physical security through a combination of security personnel, card restricted access, video surveillance, and other methods [27]. Thus, we assume that primary users are physically protected and any unauthorized entity cannot be physically close to a primary user due to those physical protection methods. We assume that secondary users are equipped with wireless radio devices and are allowed to transmit signals on the channels allocated to primary users only when the primary users are not transmitting.

We assume that an attacker's objective is to prevent other secondary users from using the primary users' channel and get an unfair share of the bandwidth when the primary users are not transmitting. Jamming attacks, which affect other users as well as the attackers themselves, are thus not in the scope of this paper.

We assume that attackers can mimic a primary user's signal and inject their fake signals into the primary user's channel. We assume that an attacker has the following capabilities: (1) He knows the signal feature of a primary user and is able to generate a fake signal with the same feature. (2) He can transmit signals on the a primary user's channel to mislead the primary user detection process at secondary users. (3) He has a large maximum transmit power that can be several times of that of a primary user. However, we assume that an attacker cannot be physically close to a primary user due to the physical protection.

We assume all secondary users have reliable ways to obtain the public key of each helper node, and an attacker cannot compromise the helper node.

IV. Overview

Our goal is to provide secondary users with the ability to determine whether a received signal is from a primary user or not in the presence of attackers. One possibility is to use the link signature of the received signal. However, as discussed in the Introduction, it is non-trivial for any secondary user to obtain the historical link signatures of a primary user in an authenticated way, given FCC's restriction on (no modification of) primary users.

In this paper, we develop a novel approach that integrates traditional cryptographic signatures and link signatures to enable primary user detection in the presence of attackers. Specifically, we propose to place a *helper node* close (and physically bound) to each primary user. Given the FCC requirement on the physical security of primary users such as TV stations, such helper nodes can also be physically protected. Though we cannot modify any primary user due to the FCC constraint, we do have the flexibility to put necessary mechanisms on each helper node, including the use of cryptographic signatures. Moreover, since each helper node is placed physically close to the primary user, their link signatures observed by a secondary user are very similar to each other.

To enable secondary users to authenticate signals from a primary user, we propose to use the helper node associated with the primary user as a "bridge". Specifically, we propose to have the helper node transmit messages when the target channel is vacant. These messages include cryptographic signatures, which will allow secondary users to verify their authenticity. As a result, secondary users can authenticate messages from the help node, then obtain the helper node's authentic link signatures, and finally verify the primary user's link signatures using those learned from the helper node. Note that our approach does not require any change to primary users, and thus follows the FCC constraint properly.

Issues of spacing multiple independent radio wave transmitters very close to each other (e.g., on the same mast) have been explored and demonstrated feasible [3], [19]. These techniques can be readily adopted to facilitate the deployment of helper nodes close to primary users in CRNs.

For the sake of presentation, in this paper, we focus our discussion on one primary user and its associated helper node. However, all discussion in this paper applies to the situations where there are multiple primary users and helper nodes, as long as the association of each primary user and its helper node is clear.

A. Technical Challenges

Two technical problems need to be resolved to make the proposed approach work. First, the helper node has to have a reliable way to detect primary user's signals. In particular, the attacker may target at the helper node. Note that the proposed approach requires that the helper node transmit messages to secondary users so that the secondary users can obtain valid *training link signatures*. However, the attacker may pretend to be the primary user and inject fake signals into the target channel. This can effectively stop the helper node, and the proposed approach will fail. Thus, it is critical for the helper node to distinguish signals from the primary users and those from the attacker.

At first glance, this seems to be the same problem as what we are trying to solve, and thus put us in a "chicken-first or egg-first" situation. However, we will show that this is not the case due to the proximity of the helper node to the primary user. We will develop a novel physical-layer authentication technique to enable the helper node to properly authenticate messages from the primary users *without using any training link signatures*. This is dramatically different from traditional link signatures, where training is a necessary part of the scheme. The details will be discussed in Section V.

Second, the interaction between the helper node and secondary users must be properly protected with lightweight mechanisms. In particular, the integration of cryptographic signatures and link signatures is a critical component of the proposed approach, and must be done efficiently. Moreover, there has to be a mechanism to prevent the attacker from replaying messages originally sent by the helper node. Otherwise, the attacker may simply reuse the valid cryptographic signatures to mislead secondary users into accepting invalid training link signatures. We will discuss critical design issues for the protocol between the helper node and a secondary user in Section VI.

V. Authenticating Primary User's Signal at the Helper Node

As discussed earlier, the helper node transmits signals using the channels allocated to its primary user such that secondary users can "learn" the link signatures of the primary user. To avoid interfering with the transmission of the primary user, the helper node transmits signals to secondary users only when the primary user is not transmitting. Therefore, the helper node should first sense the channel to decide whether the primary user is transmitting.

Unfortunately, the helper node cannot simply employ traditional primary detection approaches to determine the presence of the primary user's signal, since the attacker may mimic the primary user's signal and inject fake signals into the target channels.

In this section, we propose a novel physical-layer authentication approach that enables the helper node to authenticate the primary user's signal *without using any training link signatures*. Intuitively, the multipath effect exhibited by the primary user's signal and observed by the helper node has some unique properties, since the primary user is very close to the helper node. In our approach, we utilize such unique multipath effect to enable the helper node to distinguish the primary user's signal from those transmitted by attackers.

In the following, we first give our observation behind our new technique, then describe the proposed authentication approach, and analyze the effectiveness of the proposed approach.

A. Observation

Ideally, signal strength decreases as the signal propagates farther away from the transmitter. A short propagation path results in a large received signal amplitude, whereas a long propagation path leads to a small received signal amplitude.

Assume that there is no obstacle between the primary user and the helper node. The primary user is close to the helper node. This means the first received multipath component travels on a very short path, which is a straight line between the primary user and the helper node. Unlike the first received multipath component, the second received multipath component travels along a longer path. According to [13], the length of the path over which the second received (resolvable) multipath component travels should be larger than $\frac{c}{R}$, where c is the speed of light and R is the transmission rate.

If the distance between the primary user and the helper node is much smaller than $\frac{c}{R}$, then the amplitude of the first received multipath component is much larger than that of the second received multipath component. In other words, the amplitude ratio of the first received multipath component to that of the second received multipath component is a large number, as illustrated in Figure 3.

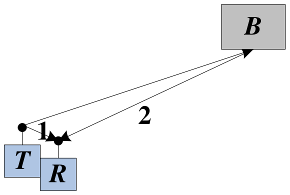


Figure 3. Amplitude ratio. T, R, and B is the primary user, the helper node, and an obstacle, respectively. The signal transmitted by T travels along two paths: path 1 ($T \rightarrow R$) and path 2 ($T \rightarrow B \rightarrow R$). Let $P1$ and $P2$ denote the amplitudes of the signal received from path 1 and path 2, respectively. The length of path 1 is much smaller than that of path 2, resulting in a large amplitude ratio $\frac{P_1}{P_2}$.

B. Authentication Method

Based on the above observation, we propose to use the amplitude ratio $r = \frac{P_1}{P_2}$ to authenticate the signal from the primary user, where P_1 and P_2 are the amplitude of the first and the second received multipath components, respectively. For each newly received signal, the helper node computes the amplitude ratio r, and then compares r with a threshold w. If $r > w$, then the received signal is marked as the primary user's signal. Otherwise, the received signal is a suspicious signal that may have been sent by an attacker, and are discarded.

For the sake of presentation, we use r_a and r_p denote the amplitude ratio of the attacker' signal and that of the primary user's signal, respectively. We would like to point out that the values of r_a and r_p depend on the positions of obstacles. Due to the randomness and uncertainty of the surroundings, r_a (r_p) may not always be smaller (larger) than the pre-determined threshold w. Hence, we may have two types of possible errors: *false alarm* and *false negative*. With a false alarm, $r_p < w$, and thus the primary user's signal is incorrectly identified as the attacker's signal. With a false negative, $r_a > w$, and thus the attacker's signal is incorrectly identified as the primary user's signal.

In Sections V-C and VII, through both theoretical analysis and experiment evaluation, we will show that the probability of false alarm and the probability of false negative decrease quickly as the distance between the attacker and the helper node increases.

1) Computing the Amplitude Ratio: A helper node can first measure the channel impulse response of a received signal, and then calculate the amplitude ratio based on the measured channel impulse response. In Lemma 1, we show that the amplitude ratio of the first multipath component to the second multipath component indeed equals the amplitude ratio of h_1 to h_2, where h_1 and h_2 are the component responses for the first and the second multipath components, respectively.

Lemma 1: Let s_1 and s_2 denote the first and the second received multipath components. The amplitude ratio r of s_1 to s_2 equals to that of h_1 to h_2, where h_1 and h_2 are the component responses for s_1 and s_2.

Proof: Recall that the channel impulse response $h(\tau)$ is $h(\tau) = \sum_{l=1}^{L} a_l e^{j\phi_l}\delta(\tau - \tau_l)$. Assume the first and the second multipath component arrives at time τ_1 and τ_2. Thus, the component responses h_1 and h_2 for the first and the second multipath components are: $h_1 = h(\tau_1) = a_1 e^{j\phi_1}\delta(0)$ and $h_2 = h(\tau_2) = a_2 e^{j\phi_2}\delta(0)$. According to [14], the amplitude ratio of h_1 and h_2 can be transformed as follows:

$$\frac{\|h_1\|}{\|h_2\|} = \frac{\|a_1 e^{j\phi_1}\delta(0)\|}{\|a_2 e^{j\phi_2}\delta(0)\|} = \frac{\|a_1(\cos\phi_1 + i\sin\phi_1)\|}{\|a_2(\cos\phi_2 + i\sin\phi_2)\|} = \frac{\|a_1\|}{\|a_2\|}$$

The channel gain a_l of the l-th multipath component is $a_l = \frac{s_l}{s_t}$, where s_l and s_t is the l-th received multipath component and the transmitted signal [14]. Therefore,

$$\frac{\|h_1\|}{\|h_2\|} = \frac{\|a_1\|}{\|a_2\|} = \frac{\|s_1\|}{\|s_2\|} = r.$$

■

Figure 4 shows a channel impulse responses obtained from the CRAWDAD data set [32], which contains over 9,300 channel impulse responses measured in an indoor environment with obstacles (e.g., cubicle offices and furniture) and scatters (e.g., windows and doors). The second multipath component arrives at the receiver about 100 microseconds after the arrival of the first one. Each multipath component leads to a triangle in shape with a peak (i.e., the component response) [23], and the helper node can use the first and the second peaks as $\|h_1\|$ and $\|h_2\|$ to compute the ratio r.

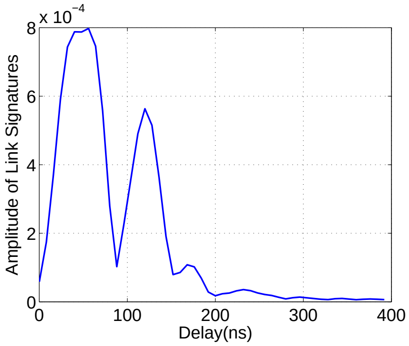


Figure 4. Computing the ratio r. This graph plots the amplitudes of a real measured channel impulse response (i.e., link signature) obtained from CRAWDAD for a 2.4 GHz channel, and $\|h_1\|$ and $\|h_2\|$ corresponds the first and the second rounded peak. Therefore, $\|h_1\| \approx 0.82 \times 10^{-3}$, $\|h_2\| \approx 0.55 \times 10^{-3}$, and $r = \frac{\|h_1\|}{\|h_2\|} \approx 1.49$.

2) Real-world Examples: Figures 5 and 6 show two real-world examples of channel impulse responses obtained from the CRAWDAD data set [32]. In Figure 5, the receiver is positioned 13.77 meters away from the transmitter. We can see that the corresponding amplitude ratio of the first multipath component to that of the second one is about $\frac{4}{2} = 2$. In Figure 6, the receiver is moved to a closer location that is 1.45 meters away from the transmitter. Now the amplitude ratio becomes $\frac{7}{0.5} = 14$.

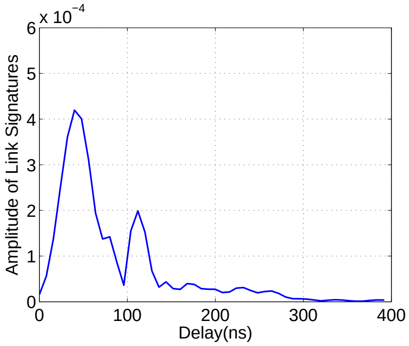


Figure 5. Example of amplitude ratio: The distance between the transmitter and the receiver is 13.77 meters, and the corresponding amplitude ratio is about about $\frac{4}{2} = 2$.

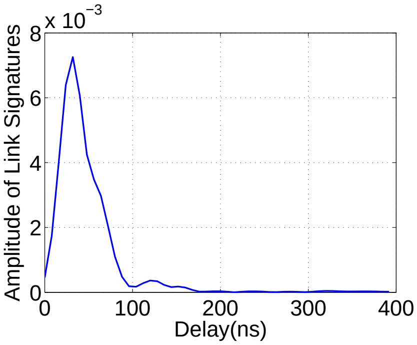


Figure 6. Example of amplitude ratio: The distance between the transmitter and the receiver is 1.45 meters, and the corresponding amplitude ratio is about $\frac{7}{0.5} = 14$.

C. Theoretical Analysis

In this section, we first give the mathematical model of the received signal amplitude, and then show the performance of the proposed authentication approach in terms of the probability of false negative (i.e., the attacker's signal is incorrectly identified as the primary user's signal) and the probability of false alarm (i.e., the primary user's signal is incorrectly identified as the attacker's signal).

1) Signal Amplitude Model: According to the simplified path loss model [14], the amplitude P_r of a received signal can be modeled as

$$P_r = \begin{cases} \sqrt{P_t k (\frac{d_0}{d})^\gamma} & d > d_0, \\ \sqrt{P_t k} & d \leq d_0, \end{cases} \tag{2}$$

where P_t is the transmit power, d is the length of the path along which the signal propagates from the transmitter to the receiver ($d > d_0$), k is a scaling factor whose value depends on the antenna characteristics and the average channel attenuation, d_0 is a reference distance for the antenna far-field, and γ is the path loss exponent. The values of k, d_0, and γ can be obtained either analytically or empirically [14].

2) Mathematical Analysis: We derive the probability of false negative and the probability of false alarm in Lemmas 2 and 3, respectively.

Lemma 2: (Probability of false negative) Given a detection threshold w, the probability p_d that the attacker's signal is wrongly identified as the primary user's signal is

$$p_d = \frac{1}{2}(1 - erf(\frac{10 \log \frac{(w^{\frac{2}{\gamma}} - 1)\sqrt{d}}{T_1 c}}{\sigma\sqrt{2}})), \tag{3}$$

where erf is the Error Function, d is the distance between the attacker and the helper node, c is the propagation speed of electromagnetic wave, and σ and T_1 are parameters that typically range between $2 - 6$dB and $0.1 - 1$ microsecond, respectively.

Proof: Let d_{a1} and d_{a2} be the lengths of the path along which the first and the second received multipath components of the attacker travels, respectively. Let P_{ra1} and P_{ra2} be the amplitudes of the first and the second multipath components, respectively. Assume $d_{a1} > d_0$ and $d_{a2} > d_0$. Thus, according to Equation 2, P_{ra1} and P_{ra1} can be approximated by

$$P_{ra1} = \sqrt{P_{ta}k(\frac{d_0}{d_{a1}})^{\gamma}},$$

$$P_{ra2} = \sqrt{P_{ta}k(\frac{d_0}{d_{a2}})^{\gamma}},$$

where P_{ta} is the transmit power of the attacker. Hence, the ratio r_a of P_{ra1} to P_{ra2} can be written as

$$r_a = \frac{\sqrt{P_{ta}k(\frac{d_0}{d_{a1}})^{\gamma}}}{\sqrt{P_{ta}k(\frac{d_0}{d_{a2}})^{\gamma}}} = \sqrt{(\frac{d_{a2}}{d_{a1}})^{\gamma}}.$$

The attacker's signal is wrongly identified as the primary user's signal if $r_a \geq w$. Thus, $p_d = 1 - \mathbb{P}(r_a < w)$. Let t_a denote the time at which the attacker's signal starts to propagate to the helper node. Let t_{a1} and t_{a2} denote the arrival times of the first and the second multipath components of the attacker, respectively. Therefore, $d_{a1} = (t_{a1} - t_a)c$ and $d_{a2} = (t_{a2} - t_a)c$, and we can have the following:

$$\begin{aligned} d_{a2} &= (t_{a2} - t_a)c \\ &= (t_{a1} - t_a)c + (t_{a2} - t_{a1})c = d_{a1} + \Delta c, \end{aligned}$$

where $\Delta = t_{a2} - t_{a1}$. According to [15], for urban, suburban, and rural areas, Δ can be statistically modeled as

$$\Delta = T_1\sqrt{d}y,$$

where T_1 is the median value of Δ when $d = 1000$m (T_1 typically ranges from $0.1 - 1$ microsecond), and y is a lognormal variate. Specifically, $Y = 10\log y$ is a Gaussian random with zero mean and a standard deviation that lies between $2 - 6$dB. The model parameters and their values can be found in Table III of [15]. Assume the first received multipath component travels along the straight line between the attacker and the helper node. Thus, $d_{a1} = d$ and

$$d_{a2} = (d + \Delta c) = d + T_1\sqrt{d}yc$$

Therefore,

$$r_a = \sqrt{(\frac{d_{a2}}{d_{a1}})^{\gamma}} = \sqrt{(\frac{d + T_1\sqrt{d}yc}{d})^{\gamma}}$$

Recall that $Y = 10\log y$ is a Gaussian random with zero mean. Let σ denote the deviation for Y. Thus,

$$\begin{aligned} p_d &= \mathbb{P}(r_a \geq w) = 1 - \mathbb{P}(r_a < w) \\ &= 1 - \mathbb{P}(\sqrt{(\frac{d + T_1\sqrt{d}yc}{d})^{\gamma}} < w) \\ &= 1 - \mathbb{P}(Y < 10\log\frac{(w^{\frac{2}{\gamma}} - 1)\sqrt{d}}{T_1 c}) \\ &= \frac{1}{2}(1 - erf(\frac{10\log\frac{(w^{\frac{2}{\gamma}}-1)\sqrt{d}}{T_1 c}}{\sigma\sqrt{2}})) \end{aligned}$$

■

Lemma 3: (Probability of false alarm) Given a detection threshold w, the probability p_f that the primary user's signal is wrongly identified as the attacker's signal is

$$p_f = \frac{1}{2}(1 + erf(\frac{10\log\frac{(w^{\frac{2}{\gamma}}-1)\sqrt{d_{p1}}}{T_1 c}}{\sigma\sqrt{2}})). \quad (4)$$

Proof: Let d_{p1} and d_{p2} be the path lengths corresponding to the first multipath component and the second multipath component of the primary user, respectively. Note that the helper node and the primary user are very close to each other. Thus, we assume that $d_{p1} \leq d_0$. Similar to d_{a1}, we assume that $d_{p2} > d_0$. Let P_{rp1} and P_{rp2} be the amplitudes of the first and the second multipath components of the primary user, respectively. According to Equation 2, P_{rp1} and P_{rp2} can be modeled by

$$P_{rp1} = \sqrt{P_{tp}k},$$

$$P_{rp2} = \sqrt{P_{tp}k(\frac{d_0}{d_{p2}})^{\gamma}},$$

where P_{tp} is the transmit power of the primary user. Hence, the ratio r_p of P_{rp1} to P_{rp2} can be written as

$$r_p = \frac{\sqrt{P_{tp}k}}{\sqrt{P_{tp}k(\frac{d_0}{d_{p2}})^{\gamma}}} = \sqrt{(\frac{d_{p2}}{d_0})^{\gamma}}.$$

The primary user's signal is wrongly identified as an attacker's signal if $r_p < w$. Thus, the probability p_f that the primary user's signal is rejected is $\mathbb{P}(r_p < w)$. Let t_p denote the time at which the primary user's signal starts to propagate to the helper node. Let t_{p1} and t_{p2} denote the arrival times of the first and the second multipath components of the primary user, respectively. Therefore,

$$d_{p2} = (t_{p2} - t_p)c = d_{p1} + T_1\sqrt{d_{p1}}yc.$$

Without loss of generality, we assume $d_{p1} = d_0$ to simplify the calculation, and thus obtain

$$r_p = \sqrt{(\frac{d_{p2}}{d_0})^{\gamma}} = \sqrt{(1 + \frac{T_1 yc}{\sqrt{d_{p1}}})^{\gamma}}.$$

Thus, we can obtain the probability p_f that the primary user's signal is wrongly identified as the attacker's signal, and

$$p_f = \mathbb{P}(r_p < w) = \mathbb{P}(\sqrt{(1 + \frac{T_1 yc}{\sqrt{d_{p1}}})^{\gamma}} < w)$$
$$= \frac{1}{2}(1 + erf(\frac{10 \log \frac{(w^{\frac{2}{\gamma}} - 1)\sqrt{d_{p1}}}{T_1 c}}{\sigma\sqrt{2}})).$$

■

3) Determining the Threshold w: The threshold w can be determined based on the requirement for the probability of false negative p_d or the probability of false alarm p_f. For practical applications, the IEEE 802.22 standard suggests both probabilities of false negative and false alarm be less than 0.1 in terms of detecting primary users [7]. Herein, we assume a stricter requirement that $p_d \leq 0.05$, and thus

$$p_d = \frac{1}{2}(1 - erf(\frac{10 \log \frac{(w^{\frac{2}{\gamma}} - 1)\sqrt{d}}{T_1 c}}{\sigma\sqrt{2}})) \leq 0.05$$

By treating w as an unknown and solve the inequality, we can get that

$$w \geq \sqrt{(1 + \frac{T_1 c \times 10^{0.11 \times \sqrt{2}\sigma}}{\sqrt{d}})^{\gamma}}. \quad (5)$$

Although the helper node does not know the actual distance d between itself and the attacker, the helper node can estimate the minimum distance d_{min} from the attacker to him/her based on the physical protection policy and the approaches he/she uses. Let

$$w_{min} = \sqrt{(1 + \frac{T_1 c \times 10^{0.11 \times \sqrt{2}\sigma}}{\sqrt{d_{min}}})^{\gamma}}.$$

w_{min} can be used as the threshold w, since Equation (5) holds when $w = w_{min}$. Note that the primary user and the helper node are very close to each other. Thus, we substitute $d_{p1} = 1$ and $w = w_{min}$ into Equation (4) and we get

$$p_f = \frac{1}{2}(1 + erf(\frac{10 \log \frac{10^{0.11 \times \sqrt{2}\sigma}}{\sqrt{d_{min}}}}{\sigma\sqrt{2}})).$$

Figure 7 shows that the probability p_f of false alarm decreases dramatically as the minimum distance d_{min} from the attacker to the helper increases. In particular, if the minimum distance is larger than 90 meters, the probability of false alarm is smaller than 0.05 for a constant 0.05 probability of false negative.

If we assume that $p_f < 0.05$, we can use the same method to get the threshold w and the corresponding probability p_d of false negative:

$$w = \sqrt{(1 + T_1 c \times 10^{-0.11 \times \sqrt{2}\sigma})^{\gamma}}.$$

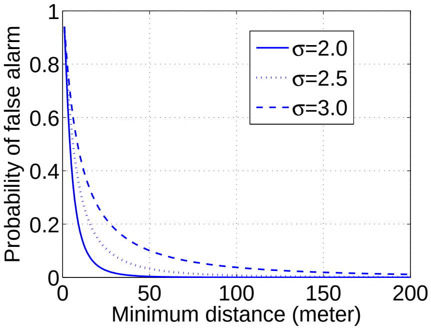


Figure 7. Probability of false alarm vs minimum distance from the attacker to the helper node for a constant 0.05 probability of false negative.

$$p_d = \frac{1}{2}(1 - erf(\frac{10 \log 10^{-0.11 \times \sqrt{2}\sigma} \sqrt{d}}{\sigma\sqrt{2}})).$$

Figure 8 shows the probability of false negative for a constant 0.05 probability of false alarm.

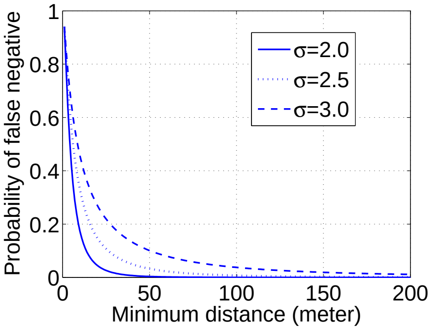


Figure 8. Probability of false negative vs minimum distance from the attacker to the helper node for a constant 0.05 probability of false alarm.

Figure 9 displays the tradeoff between the probability of false alarm and the probability of false negative, when $\sigma = 2.5$ and the minimum distance between the attacker and the helper node is 50, 60, and 70 meters, respectively.

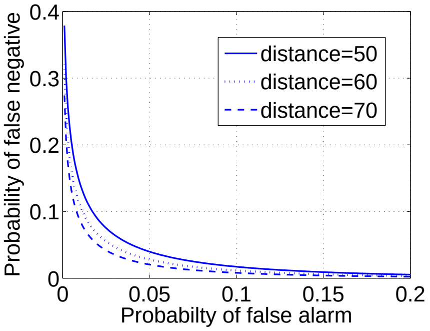


Figure 9. Tradeoff between probability of false alarm and the probability of false negative.

VI. Interaction between the Helper Node and Secondary Users

Intuitively, a helper node can notify secondary users if the channel is open to them, since the helper node itself has the ability to authenticate a primary user's signal. However, in this paper, we utilize link signatures to let secondary users identify a primary user's signal in a proactive way even when the helper node is sleeping.

The objective of having a secondary user interact with the helper node is to allow the secondary user to learn valid link signature from the helper node. Thus, the interaction between the secondary user and the helper node can be considered as a *training process*, during which the secondary user collects enough valid link signatures that could be used to verify future signals from the primary user. For convenience, we refer to the link signatures collected during the training process as *training link signatures*, and the packets from the helper node as *training packets*.

Note that the helper node is not required to transmit training packets all the time, and the training process may be triggered periodically or at the requests of secondary users. Our scheme allows the helper node to sleep during non-training period (e.g., the time interval between the end of a training process and the beginning of the subsequent training process). However, secondary users can still work in a proactive way even when the helper node is sleeping. As a result, the probability of interfering the transmission of the primary user is reduced. With training link signatures acquired in training processes, secondary users can directly verify whether a newly received signal is from the primary user or not.

A. Obtaining Training Link Signatures

We assume that the helper node is able to deliver training packets to secondary users. For example, the helper node may periodically sense the channel and broadcast training packets to all secondary users if the channel is open. Alternatively, we may use a request/reply protocol between secondary users and the helper node. In other words, if a secondary user does not have enough training link signatures, it sends a request to the helper node through the control channel, and the helper node then transmits training packets back upon request. Our approach is independent of the exact way training packets are triggered.

Upon receiving a packet from the helper node, a secondary user measures the link signature and verifies the cryptographic signature in the received packet. If the cryptographic signature is valid, the secondary user accepts the corresponding link signature. Otherwise, the secondary user has to discard both the link signature and the received packet.

It is well-known that public key cryptographic signatures are expensive to generate and verify. A straightforward application of cryptographic signatures will lead to substantial overheads on the helper node as well as secondary nodes. To enable efficient interaction between the helper node and a secondary user, we propose to amortize the signature generation and verification costs on both helper node and secondary users.

Note that there are known ways for signature amortization using cryptographic hash functions (e.g., [24], [31]). Thus, we consider our contribution here secondary (compared with the authentication method in Section V).

Amortizing Cryptographic Signature Costs: The helper node randomly picks a number r_l and uses a one-way cryptographic hash function H to generates a one-way hash chain $r_0 \leftarrow r_1 \leftarrow ... \leftarrow r_l$, where $r_{i-1} = H(r_i)$ for $1 \leq i \leq l$. It is well-known that given an authenticated value r_i in this hash chain, it is easy to authenticate any later value r_j $(i < j < l)$. However, it is computationally infeasible to derive any later r_j $(i < j < l)$ if no value beyond r_i is known.

To reduce the signature cost, for each hash chain, the helper node generates one and only one cryptographic signature on r_0 using its private key. Let $sig(r_0)$ denote the signature. Suppose the helper node needs to authenticate the i-th packet since the generation of the hash chain. The helper node then place r_0, $sig(r_0)$, i, and r_i in the packet. (The helper node should certainly start with $i = 0$.) Thus, the helper node never needs to generate another signature for this hash chain.

Consider a secondary node that receives a packet using the above hash chain from the helper node for the first time. Note that i could be greater than 0 if this secondary node has not received any packet from the helper node recently. The secondary node then first verifies the signature $sig(r_0)$. If $i = 0$, the secondary node has successfully verified the cryptographic signature from the helper node. However, if $i > 0$, the secondary node needs to future hash r_i for i times and compare $H^{<i>}(r_i)$ with r_0. If they match, the packet is also valid. In any case, the secondary node should save r_i for future authentication.

If the secondary node has received and verified a signature from the helper node with the same hash chain previously, it must have saved an authenticated hash value r_j $(j < i)$. As a result, the secondary node does not have to verify the signature $sig(r_0)$ again. Instead, it only needs to compute $H^{<i-j>}(r_j)$ and compare the result with r_i. A match indicates a successful authentication of the packet.

As we can see, the helper node needs to generate one and only one cryptographic signature for each hash chain. Similarly, each secondary node only needs to verify one cryptographic signature once for each hash chain. Thus, this amortization approach can greatly reduce the computational overheads on both the helper node and the secondary node.

Defending against Replay Attacks: As discussed earlier, a critical threat is that the attacker may replay intercepted training packets from a valid helper node at its own location. As a result, the attacker can convince secondary users to accept the attacker's link signatures as training link signatures. Since the secondary users are not guaranteed to have received the original transmission, traditional anti-replay mechanisms such as sequence numbers, which are intended for detecting replayed packet contents (rather than replayed signals), will not work.

Fortunately, there are multiple known techniques to handle replayed signals in wireless networks, such as the hardware-based, authenticated Medium Access Control (MAC) layer timestamping [33] and the method for detecting wireless signals tunneled by a malicious node [20]. These techniques can be adopted in CRNs to enable a secondary node and the helper node to detect replayed training packets.

Alternatively, we may take advantage of potentially synchronized clocks between valid secondary users and the helper node to defend against such threats. According to IEEE 802.22 standard [9], secondary users and base stations are "required to use satellite based geo-location technology, which will also facilitate synchronization among neighboring networks by providing a global time source." We can assign each value in the above hash chain to a specific point in time. These times can be pre-scheduled such that all secondary users know when each hash value should be used. The helper node then transmits each hash value at the pre-scheduled point in time, provided that the primary user is not using the channel. When a secondary user receives a training packet, it can use its local time and the pre-scheduled time to estimate the transmission time of this packet. An overly long transmission time indicates that the packet has been replayed by the attacker.

Learning Training Link Signatures: To compute the training link signature, the secondary user samples the received signal using an A/D sampler, stores the first $\kappa+1$ samples in a buffer, and demodulates the samples of the received signal into a packet. If the packet can pass authentication, the secondary user computes the link signature of the packet using the stored $\kappa + 1$ samples. (See the Appendix for how to compute link signatures.) Otherwise, the secondary user discards the stored samples. The secondary user typically needs to obtain a series of training link signatures for verifying future signals.

B. *Verifying Link Signatures*

For a newly received signal s_N, the secondary user first measures its link signature, which is denoted by $\mathbf{h}^{(N)}$, and then use training link signatures to verify $\mathbf{h}^{(N)}$.

Let $\mathcal{H} = \{\mathbf{h}^{(n)}\}_{n=1}^{N-1}$ denote the set of training link signatures, where $\mathbf{h}^{(n)}$ is the link signature measured from the i-th received training packet. The secondary user can verify whether s_N is transmitted by the primary user or not using the location distinction algorithm proposed in [23]. Specifically, the secondary user calculates the distance (i.e., difference) between $\mathbf{h}^N$ and the training set $\mathcal{H}$, and then compares the distance with a threshold. If the distance is less than a threshold, s_N is marked as the primary user's signal. Otherwise, s_N may be sent by the attacker and the secondary user ignores it. The method that can be used to calculate distance is discussed in [23].

VII. Experimental Evaluation

Our approach involves two types of authentication: authentication of the primary user's signal at the helper node, and authentication of the primary user's signal at a secondary user. In this section, we report our experimental evaluation to show the effectiveness of both methods.

We validate the proposed authentication methods using the CRAWDAD data set [22], which includes over 9,300 real channel impulse response measurements (i.e., link signatures) in a 44-node wireless network [32]. There are $44 \times 43 = 1,892$ pairwise links between the nodes, and multiple measurements are provided for each link [32]. The map of the 44 node locations is shown in [23]. The measurement environment is an indoor environment with obstacles (e.g., cubicle offices and furniture) and scatters (e.g., windows and doors). More information regarding the CRAWDAD data set can be found in [22], [32].

A. *Authentication at the Helper Node*

To avoid interfering with the primary user's transmission, the helper node needs to first sense the channel, and verify whether a received signal is from the primary user. As discussed earlier, false alarms and false negatives may occur during the authentication process. Thus, we evaluate the performance of the authentication method in terms of the probability of false negative and the probability of false alarm.

Recall that during authentication the helper node computes the amplitude ratio of the first multipath component to the second multipath component for each received signal. If the amplitude ratio is larger than a threshold, then the received signal is considered from the primary user. Otherwise, it is considered from the attacker. Hence, false alarms happen when the primary user's amplitude ratio is less than the threshold, and false negative happens when the attacker's amplitude ratio is larger than the threshold.

1) Probability of False Alarm: To obtain the amplitude ratio of the primary user's signal, we perform experiments as follows. For $1 \leq i \leq 44$, we assume that node i is the helper node. For each of the remaining nodes, if there is no obstruction between itself and node i, we mark it as a *line-of-sight node*. Among all line-of-sight nodes for node i, we pick the one that is closest to node i as an approximation of the primary user p. Note that some nodes do not have line-of-sight nodes in their vicinities, and thus they are not used in our experiment (e.g., nodes 8 and 29 in the map shown by [23]). Finally, we compute the amplitude ratio using the primary user's channel impulse responses (i.e., link signatures of link (p, i)). The CRAWDAD data set has multiple measurements for each link. Thus, we can get multiple amplitude ratios for each link. We sort the collected amplitude ratios and compute empirical cumulative distribution function (CDF) for them. Let N denote the number of the collected amplitude ratios, $F(x)$ denote the empirical CDF, and $x_1, ..., x_N$ denote the sorted amplitude ratios, where $x_i \leq x_j$ for $1 \leq i \leq j \leq N$. The empirical CDF $F(x_i)$ is given by $F(x_i) = \frac{n_{\leq x_i}}{N}$, where $n_{\leq x_i}$ is the number of amplitude ratios that are less than or equal to x_i.

Figure 10 shows the empirical CDF curve of the amplitude ratios computed using primary users' channel impulse responses. This CDF curve can be used to derive the probability of false alarm directly. For example, about 5% amplitude ratios are less than or equal to 5. Hence, if the threshold is set to 5, then 5% amplitude ratios are smaller than the threshold and the probability of false alarm is 0.05.

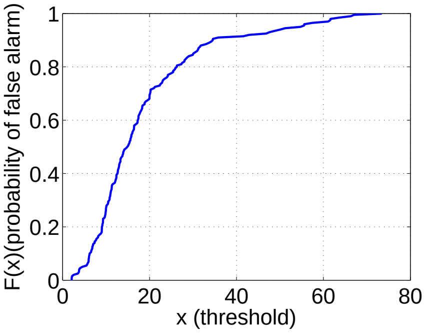


Figure 10. The empirical CDF curve of amplitude ratios computed using primary users' channel impulse responses

2) *Probability of False Negative:* We perform experiment to examine the amplitude ratios of attackers' signals. For $1 \leq i \leq 44$, we assume that node i is the helper node and find its primary user p using the same method as discussed in the above experiment. For each of the remaining nodes, we calculate the distance between this node and the helper node. We mark the node as the attacker if the calculated distance is larger than r times of the distance between the helper node and the primary user, where r is set to 2, 4, and 8 in our experiment. We compute the amplitude ratio for node i using the attacker's channel impulse responses (i.e., link signatures of link (a, i), where a is the node index of the attacker).

Figure 11 shows the empirical CDF curves of all amplitude ratios computed using attackers' channel impulse responses. In particular, about 95% amplitude ratios of attackers' signals are less than or equal to 5 for all possible values of r (i.e., $r = 2, 4, 8$). Based on the empirical CDF of the amplitude ratios, we generate Figure 14 to show the relationship between the probability of false negative and the threshold. For instance, the empirical CDF indicates that about 95% amplitude ratios are less than or equal to 5. Hence, about 5% amplitude ratios are larger than 5 and the probability of false negative is 0.05 if the threshold is set to 5. It is shown in Figure 14 that the probability of false negative decreases as the distance between the attacker and the helper node increases (i.e., r gets larger).

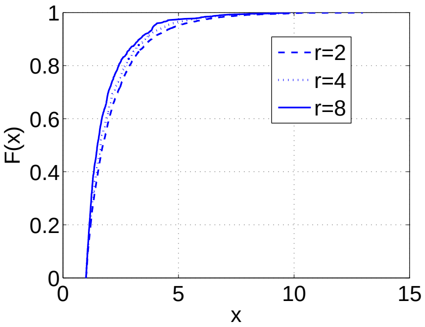


Figure 11. CDF curves of amplitude ratios computed using attackers' link signatures.

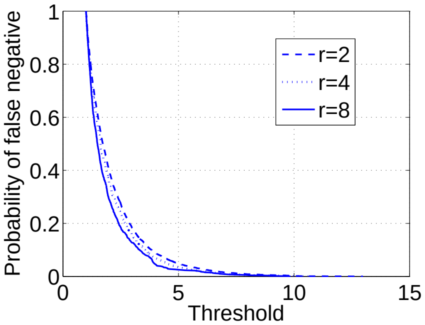


Figure 12. Probability of false negative vs threshold

3) *Trade off between Probability of False Alarm and Probability of False Negative:* Let P_{FA} and P_{FN} denote the probability of false alarm and false negative, respectively. We analyze the trade off between P_{FA} and P_{FN} by examining the relationship between P_{FA} and the threshold, as well as the relationship between P_{FN} and the threshold. For a particular value of threshold, the authentication approach would achieve a particular P_{FA} and P_{FN}.

Table I shows the probability P_{FN} when the probability P_{FA} of false alarm ranges between 0.05 and 0.2. If $P_{FA} = 0.05$, P_{FN} is less than 0.0655, 0.0486, and 0.0321 for $r = 2$, 4, and 8, respectively. For a constant P_{FA}, P_{FN} decreases as the distance between the attacker and the helper node increases (i.e., r increases). In particular, $P_{FN} = 0.0655$ when the distance between the attacker and the helper node is larger than twice of the distance between the primary user and the helper node (i.e., $r = 2$). However, P_{FN} falls to 0.0321 when the distance between the attacker and the helper node is 8 times larger than the distance between the primary user and the helper node (i.e., $r = 8$).

Table I
TRADE OFF BETWEEN P_{FA} AND P_{FN}

P_{FA}	P_{FN} (r=2)	P_{FN} (r=4)	P_{FN} (r=8)
0.05	≤ 0.0655	≤ 0.0486	≤ 0.0321
0.1	≤ 0.0248	≤ 0.0163	≤ 0.0155
0.15	≤ 0.0109	≤ 0.0070	≤ 0.0066
0.2	≤ 0.0053	≤ 0.0032	≤ 0.0022

B. *Authentication at Secondary Users*

During the authentication process at a secondary user, the secondary user needs to verify whether a received signal is from the primary user or not by looking at the distance between the corresponding link signature and the training set. We refer to the distance as *link difference*. If the link difference is smaller than a threshold, then the received signal is considered from the primary user. Otherwise, the signal is considered to be sent by an attacker and the secondary user discards it.

Therefore, a false alarm happens if the link difference between the primary user's link signature and the secondary user's training set is larger than the threshold, and a false negative happens if the link difference between the attacker's link signature and the secondary user's training set is smaller than the threshold. Similar to the authentication at the helper node, we use the probability of false alarm and the probability of false negative to measure the performance of the proposed approach.

In our experiment, we compute the link differences between the primary user's link signature and the secondary user's training set, as well as the link differences between the attacker's link signature and the secondary user's training set. Based on their statistical distributions, we examine how likely false alarms and false negatives would happen.

1) Probability of False Alarm: To get the link differences between link signatures of the primary user and the secondary user's training set, we perform experiment as follows. We pick all nodes one by one as the primary user. Starting with node 1, we use the node closest to node 1 to approximate the helper node (i.e., node 3 in the map [23]). We further assume that all the other nodes (i.e., node 2 and nodes 4-44 on the map [23]) are secondary users. For each secondary user s, we generate its training set using all link signatures of the node pair $(3, s)$ (i.e., the helper node's link signatures) and k ($k = 0, 1, 2$) link signatures of the node pair $(1, s)$ (i.e., the primary user's link signatures). Then, we compute link differences $d_s^1, ..., d_s^{n_p}$ between the primary user's link signatures and the training set of node s, where n_p is the number of primary user's link signatures.

We use the average value of d_s^1,... , $d_s^{n_p}$ as the link differences between the link signatures of node 1 and the training sets of each secondary user s. Similarly, we assume that nodes 2,...,44 are primary users and perform the same process to get the link differences between the link signatures of nodes 2,...,44 and the training sets of the secondary users.

Figure 13 shows curves of the empirical CDFs for the collected link differences, where each training set contains all measured link signatures of a helper node, and k ($k = 0, 1, 2$) measured link signatures of a primary user. Almost all link differences are less than or equal to 10 when the training set only contains the link signatures of a helper node (i.e., $k = 0$). Once a primary user's link signature is added to the training set (i.e., $k = 1$), the link differences decreases dramatically. Figure 14 shows the relationship between the probability of false alarm and the threshold.

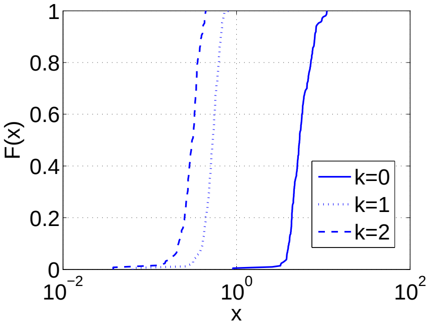


Figure 13. CDF curves of link differences between the link signatures of primary users and the training sets of secondary users.

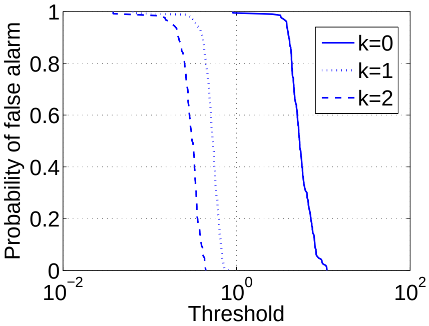


Figure 14. Probability of false alarm vs threshold

2) Probability of False Negative: We also perform experiment to examine the link differences between link signatures of attackers and training sets of secondary users. We assume that node 1 is the attacker. We pick node p as the primary user and node s as the secondary user such that $p \neq s \neq 1$. For each combination of p and s, we first find the helper node of p. Let p_h denote the helper node. If $p_h \neq s \neq 1$, we generate the training set of s using the same approach as the first experiment. We then compute the link difference $d_{s,p}^1, ..., d_{s,p}^{n_a}$ between the attacker's link signatures and the training set, where n_a is the number of attacker's link signatures. After scanning all combinations, we use the average value of $d_{s,p}^1$,..., the average value of $d_{s,p}^{n_a}$ as the link difference between link signatures of node 1 and the training sets of secondary users. Similarly, we assume that nodes 2,...,44 are attackers and perform the same process to get the link differences between the link signatures of nodes 2,...,44 and the training sets of secondary users.

Figure 15 shows the empirical CDF curves of the collected link differences for $k = 0$, $k = 1$, and $k = 2$. Note that the empirical CDF curves can be used to derive the probability of false negative directly given a threshold. For example, about 10% link differences are less than or equal to 7.5 when $k = 0$. This means the probability of false negative is 0.1 for a threshold of 2.5.

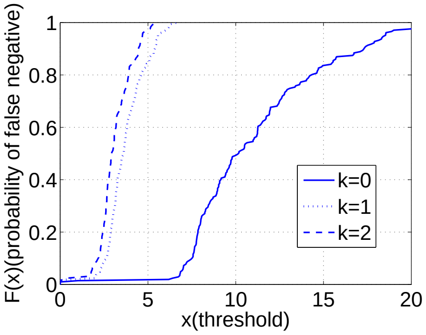


Figure 15. CDF curves of link differences between link signatures of attackers and the training sets of secondary users.

3) Trade off between Probability of False Alarm and Probability of False Negative: We derive the trade off between the probability P_{FA} of false alarm and the probability P_{FN} of false negative by analyzing the relationship between P_{FA} (P_{FN}) and thresholds. Table II shows the result. To achieve a 0.05 probability of false alarm, the probability of false negative is less than 0.3188, which is actually a loose upper bound. In our experiment, we use the node closest to the primary user to approximate the helper node; there is indeed an unnecessarily long distance between the primary user and its helper node. Thus, the probability of false negative is unnecessarily large in our experiment.

Note that all P_{FN}'s are less than the same value 0.0241 for $k = 1$. This is because the threshold ranges between 0.6282 and 0.6816 when $0.05 \leq P_{FA} \leq 0.2$. This range is quite narrow, and we can only find a single P_{FN} from the empirical CDF of the attackers' link differences. Similarly, all P_{FN}s are less than the same value 0.0240 for $k = 2$.

VIII. Implementation

We demonstrate the feasibility of the proposed approach using a prototype implementation on Universal Software Radio Peripherals (USRPs) based on GNUradio [1]. Although wireless signals transmitted by USRPs may not exhibit the multipath properties due to low bandwidths, low power, and short range communication with USRPs, the prototype implementation nevertheless demonstrates the feasibility of

Table II
Trade off between P_{FA} and P_{FN}: the probability P_D of false negative decreases as k increases

P_{FA} ($k = 0, 1, 2$)	P_{FN} (k=0)	P_{FN} (k=1)	P_{FN} (k=2)
0.05	≤ 0.3188	≤ 0.0241	≤ 0.0240
0.1	≤ 0.2319	≤ 0.0241	≤ 0.0240
0.15	≤ 0.1063	≤ 0.0241	≤ 0.0240
0.2	≤ 0.0821	≤ 0.0241	≤ 0.0240

integrating cryptographic signatures and link signatures for authenticating primary users' signals in CRNs.

A USRP is a radio frequency (RF) front end that has an analog to digital (AD) and a digital to analog (DA) converter, which can achieve an input and output sampling rate up to 64 Mb/s and 128 Mb/s, respectively. GNUradio is a software toolkit consisting of signal processing blocks that can be used to implement software radios on readily-available, low-cost external RF hardware and commodity processors (e.g., USRPs) [1].

We connect one USRP to a Lenovo X61 laptop (1.80 GHz Intel Core Duo CPU), and one USRP to a DELL machine (3.40 GHz Intel Pentium 4 CPU) via USB 2.0 links. Both computers are running Linux (Ubuntu 9.04) and GNUradio (version 3.2.2), and both USRPs employ XCVR2450 daughter boards as transceivers. We implement the helper node and the secondary user applications using GNUradio toolkit, and install the helper node and the secondary user application on the laptop and the DELL machine, respectively.

The helper node application generates signed packets using the method described in Section VI-A, where we employed MD5 as the one-way function and RSA as the cryptographic signature algorithm. The signed packets are modulated into physical layer symbols by a differential binary phase-shift keying (DBPSK) modulator. Then all physical layer symbols enter a pulse shape filter, which transforms those symbols into baseband signals. The baseband signals are delivered to the USRP, converted into RF signals, and finally transmitted to the wireless channel through the antenna.

Upon capturing a RF signal, the secondary user application down-converts the RF signal into baseband signal. Then the baseband signal is recorded and delivered to a DBPSK demodulator. If the output of the demodulator can pass the verification, the secondary user reconstructs the transmitted signal from the demodulation output, and computes the 512-points complex Fourier transform F_1 and F_2 of the baseband signal and the transmitted signal, respectively. Finally, F_1 is multiplied by the conjugate of F_2, and the inverse Fourier transformation is used to calculate the link signature as described in the Appendix.

In our experiment, the packet length is 75 bytes, the bit rate is 2Mbit/s, and the carrier frequency is 5GHz. The laptop and the Dell machine are used as the transmitter and the receiver, respectively. We first put the transmitter about 5 meters away from the receiver, and let the transmitter send a signed packet to the receiver. Upon reception of the packet, the receiver verifies the cryptographic signature in the packet and measures the link signature. Then we move the transmitter to position a and position b, which is about 0.5 meter and 15 meters away from the old position, respectively. At both positions, we let the transmitter transmit signed packets to the receiver. Figure 16 displays the measured link signatures for different positions, we observe that the link signatures of the old position and position a are mixed together, and the link signature of position b greatly deviates from the mixed ones. This observation is consistent with our analytical result.

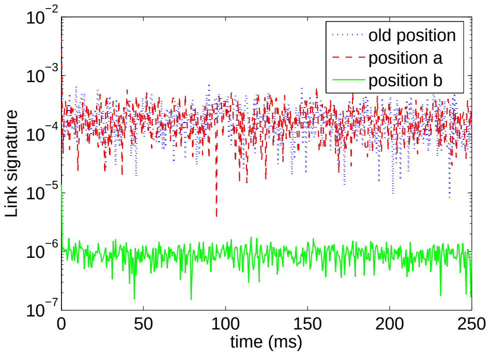


Figure 16. Measured link signatures for old position, position a, and position b

In our approach, generating signatures, verifying signatures, computing Fourier transform, and inverse Fourier transform are four major operations that are indispensable. To get an intuitive feeling of the computational overhead introduced by these operations, we did an experiment using the prototype system to test the computation time. We let the transmitter transmits 1,000 packets to the receiver every 0.1 second, and record the computation time by those operations.

Note that the transmitter (or the receiver) only needs to generate (or verify) the cryptographic signature $sig(r_0)$ in the first packet. For all the following packets, the transmitter signs them by simply appending $sig(r_0)$ and the corresponding hash values to them, and the receiver verifies them by computing and comparing hash values. Table III shows the time costs of signing (verifying) those packets. In practice, the calculation of link signatures can be performed more efficiently with Fourier transform implemented on special hardware (e.g., Virtex 2 Pro 50 Fast Fourier transform (FFT) core, which can finish the 512 points complex Fourier transform with in less than 5.5 microseconds).

IX. Related Work

Primary user detection has been intensively studied in the past few years (e.g., [12], [17], [25], [26], [29], [30], [34], [37]). Traditional detection techniques in general can be categorized into energy detection (e.g., [30]) and feature detection (e.g., [12], [25], [26], [29], [37]). In energy detection, any captured signal whose energy exceeds a threshold is identified as a primary user's signal. In feature detection, signal features (e.g., pilot, synchronization words, and cyclostationarity) are extracted and used to detect the presence of a primary user's signal. However, those traditional techniques will fail in hostile environments, where an attacker transmits with large power or mimics a primary user's signal features to gain unfair share of the bandwidth.

Table III
Computation time (milliseconds)

Operations	Time range	Average
Signing	0.1699-0.0239	0.0441
Verification	0.4519-0.0781	0.1288
Fourier transform	1.4000-0.4200	0.5612
Inverse Fourier transform	0.7310-0.21901	0.2920

A recent attempt considered the security aspects of primary detection and proposed to utilize RSS-based location distinction for detecting primary users' signals in the presence of attackers [4]. Specifically, a secondary user verifies whether a received signal is from a primary user or not by estimating the location of the signal source. If the estimation result deviates from the known location of the primary user, it is highly possible that the signal is sent by an attacker. However, as indicated in [23], RSS-based location distinction approach, which is used in [4], can be easily disrupted if the attacker is equipped with array antennas. Moreover, such an approach requires multi-node collaboration, which is expensive in terms of bandwidth and energy.

In our work, we designed a primary user detection scheme by exploiting link signatures, which do not have the weakness of RSS based location distinction and achieve a higher accuracy [23], [38]. We integrate cryptographic and wireless link signatures to authenticate a primary user's signal, and use two levels of detections (i.e., detection at a helper and at a secondary user) to address the technical challenges caused by adopting link signatures in CRNs.

There are other related works, including cooperative feature detection or energy detection [11], [21], [28], [36], secure data fusion in the presence of false information for distributed spectrum sensing [5], [35], performance evaluation of primary user detection in IEEE 802.22 [7], trade off between a secondary user's data transmission and the detection of a primary user's signal [16], and IEEE 802.22 standard for CRNs [8], [9]. These works are complementary to ours.

X. Conclusion

In this paper, we developed a novel approach for authenticating primary users' signals in CRNs, which conforms to

FCC's requirement. Our approach integrates cryptographic signatures and wireless link signatures to enable primary user detection in the presence of attackers. Essential to our approach is a *helper node* placed physically close to each primary user, which serves as a "bridge" to enable a secondary user to verify the cryptographic signature carried by the helper node's signals, and then obtain the helper node's authentic link signatures to verify the primary user's signals. A key contribution in our paper is a novel physical layer authentication technique that enables the helper node to authenticate signals from its associated primary user. Unlike previous techniques for link signatures, our approach explores the geographical proximity of the helper node to the primary user, and thus does not require any training process.

We have examined the proposed approach through theoretical analysis, experimental evaluation using the CRAWDAD data set [22], and a prototype implementation on USRPs based on GNUradio [1]. Our results indicate that the proposed approach is a promising solution for authenticating primary users' signals in CRNs.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful suggestions. The link signature data set used in this paper was provided by Motorola Labs, Florida Communications Research Lab, and the University of Utah [22]. This work is supported by the National Science Foundation (NSF) under grants CAREER-0447761 and CNS-0721424. The contents of this paper do not necessarily reflect the position or the policies of the U.S. Government.

REFERENCES

[1] Gnu radio software. http://gnuradio.org/trac.

[2] Omni antenna vs. directional antenna. https://www.cisco.com/en/US/tech/tk722/tk809/technologies_tech_note09186a00807f34d3.shtml#topic4.

[3] White-paper: 3g infrastructure sharing. *Siemens*, 2001.

[4] R. Chen, J. Park, and J. H. Reed. Defense against primary user emulation attacks in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 26(1):25–37, 2008.

[5] R. Chen, J. M. Park, and K. Bian. Robust distributed spectrum sensing in cognitive radio networks. In *Proceedings of IEEE INFOCOM 2008 mini-conference*, April 2009.

[6] Federal Communications Commission. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing spectrum agile radio technologies. *ET Docket*, (03-108), Dec. 2003.

[7] C. Cordeiro, K. Challapali, and M. Ghosh. Cognitive phy and mac layers for dynamic spectrum access and sharing of tv bands. In *TAPAS '06: Proceedings of the first international workshop on Technology and policy for accessing spectrum*, page 3, New York, NY, USA, 2006. ACM.

[8] C. M. Cordeiro, K. Challapali, and D. Birru. Ieee 802.22: An introduction to the first wireless standard based on cognitive radios. *Journal of communications*, 1, April 2006.

[9] C.Stevenson, G. Chouinard, Z. Lei, W. Hu, S. Shellhammer, and W. Caldwell. Ieee 802.22: The first cognitive radio wireless regional area network standard. *Communications Magazine, IEEE*, 47, January 2009.

[10] ETTUS. Usrp-universal software radio peripheral. http://www.ettus.com.

[11] G. Ganesan and Y. Li. Cooperative spectrum sensing in cognitive radio networks. In *Proceedings of IEEE DySPAN*, pages 137–143, November 2005.

[12] L. P. Goh, Z. Lei, and F. Chin. Dvb detector for cognitive radio. In *ICC'07: Proceedings of the International Conference on Communications 2007*, pages 6460–6465, 2007.

[13] A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[14] A. Goldsmith. *Wireless Communications*. Cambridge University Press, August 2005.

[15] L. J. Greenstein, V. Erceg, Y. S. Yeh, and M. V. Clark. A new path-gain/delay-spread propagation model for digital cellularchannels. *IEEE Transactions on Vehicular Technology*, 46:477–485, 1997.

[16] A.T. Hoang and Y.-C. Liang. Adaptive scheduling of spectrum sensing periods in cognitive radio networks. In *Proceedings of the the IEEE GLOBECOM 2007*, pages 3128–3132, November 2007.

[17] H. Kim and K. G. Shin. In-band spectrum sensing in cognitive radio networks: energy detection or feature detection? In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 14–25, New York, NY, USA, 2008. ACM.

[18] L. B. Kuechle. Selecting receiving antennas for radio tracking. http://www.atstrack.com/PDFFiles/receiverantrev6.pdf.

[19] T. Leibner. Network and infrastructure sharing in 2g networks. *Siemens*, 2004.

[20] D. Liu, P. Ning, and W.K. Du. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS '05)*, pages 609–619, June 2005.

[21] S. M. Mishra, A. Sahai, and R. Brodersen. Cooperative sensing among cognitive radios. In *ICC'06: Proceedings of the International Conference on Communications 2006*, volume 4, pages 1658–1663, 2006.

[22] N. Patwari and S. K. Kasera. CRAWDAD utah CIR measurements. http://crawdad.cs.dartmouth.edu/meta.php?name=utah/CIR.

[23] N. Patwari and S. K. Kasera. Robust location distinction using temporal link signatures. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 111–122, New York, NY, USA, 2007. ACM.

[24] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient authentication and signing of multicast streams over lossy channels. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, May 2000.

[25] Y. Qi, T. Peng, W. Wang, and R. Qian. Cyclostationarity-based spectrum sensing for wideband cognitive radio. In *CMC '09: Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, pages 107–111, Washington, DC, USA, 2009. IEEE Computer Society.

[26] A. Sahai and D. Cabric. Cyclostationary feature detection. *Tutorial presented at the IEEE DySPAN 2005 (Part II)*, November 2005.

[27] Media Security and Reliability Council. Communications infrastructurevsecurity, access, and restoration working group, final report. http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-244430A1.pdf, Feb. 2004.

[28] S. Shankar, C. Cordeiro, and K. Challapali. Spectrum agile radios: utilization and sensing architectures. In *Proceedings of IEEE DySPAN*, pages 160–169, November 2005.

[29] S. Shellhammer. An atsc detector using peak combining. *IEEE 802.22-06/0243r0*, November 2006.

[30] S. Shellhammer, S. Shankar N., R. Tandra, and J. Tomcik. Performance of power detector sensors of dtv signals in ieee 802.22 wrans. In *TAPAS '06: Proceedings of the first international workshop on Technology and policy for accessing spectrum*, New York, NY, USA, 2006. ACM.

[31] D. Song, D. Zuckerman, and J.D. Tygar. Expander graphs for digital stream authentication and robust overlay networks. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.

[32] SPAN. Measured channel impulse response data set. http://span.ece.utah.edu/pmwiki/pmwiki.php?n=Main.MeasuredCIRDataSet.

[33] K. Sun, P. Ning, C. Wang, A. Liu, and Y. Zhou. TinySeRSync: Secure and resilient time synchronization in wireless sensor networks. In *Proceedings of 13th ACM Conference on Computer and Communications Security (CCS '06)*, pages 264–277, October/November 2006.

[34] V. Tawil. Dtv signal captures. *IEEE 802.22-06/0038r0*, March 2005.

[35] W. Wang, H. Li, Y. L. Sun, and Z. Han. Attack-Proof collaborative spectrum sensing in cognitive radio networks. In *IEEE 43rd Annual Conference on Information Sciences and Systems*, 2009.

[36] B. Wild and K. Ramchandran. Detecting primary receivers for cognitive radio applications. In *Proceedings of IEEE DySPAN*, pages 124–130, November 2005.

[37] W. Xia, S. Wang, W. Liu, and W. Cheng. Correlation-based spectrum sensing in cognitive radio. In *CoRoNet '09: Proceedings of the 2009 ACM workshop on Cognitive radio networks*, pages 67–72, New York, NY, USA, 2009. ACM.

[38] J. Zhang, M. H. Firooz, N. Patwari, and S. K. Kasera. Advancing wireless link signatures for location distinction. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, New York, NY, USA, 2008. ACM.

Appendix

In this appendix, we explain how we use the methodology proposed in [23] to compute the link signature of the stored $\kappa + 1$ samples.

Let $\mathbf{r} = [r(0), ..., r(\kappa T_r)]$ denote the samples of the received signal $r(t)$, where T_r is the sampling rate. Based on the demodulation results, the secondary user can recreate the transmitted signal $s(t)$. Let $\mathbf{s} = [s(0), ..., s(\kappa T_r)]$ denote the corresponding $\kappa + 1$ samples of the transmitted signal $s(t)$. Let $R(iT_r)$ and $S(iT_r)$ be the discrete Fourier transform of $r(iT_r)$ and $s(iT_r)$, respectively. According to [23], the link signature $\mathbf{h} = [h(0), ..., h(\kappa T_r)]$, which are the $\kappa + 1$ samples of $h(t)$, can be calculated as

$$h(iT_r) = \frac{1}{\mathcal{P}_s} F^{-1}(S^*(iT_r)R(iT_r)),$$

where $F^{-1}(\cdot)$ denote the inverse discrete Fourier transform, $S^*(iT_r)$ is the complex conjugate of $S(iT_r)$, and $\mathcal{P}_s = S^*(iT_r) * S(iT_r)$.