Cryptographic Link Signatures for Spectrum Usage Authentication in Cognitive Radio*

Xi Tan, Kapil Borle, Wenliang Du and Biao Chen
Dept. of Electrical Engineering & Computer Science, Syracuse University
Syracuse, New York, USA
{xtan, kmborle, wedu, bichen}@syr.edu

ABSTRACT

It was shown that most of the radio frequency spectrum was inefficiently utilized. To fully use these spectrums, cognitive radio networks have been proposed. The idea is to allow secondary users to use a spectrum if the primary user (i.e., the legitimate owner of the spectrum) is not using it. To achieve this, secondary users should constantly monitor the usage of the spectrum to avoid interference with the primary user. However, achieving a trustworthy monitoring is not easy. A malicious secondary user who wants to gain an unfair use of a spectrum can emulate the primary user, and can thus trick the other secondary users into believing that the primary user is using the spectrum when it is not. This attack is called the Primary User Emulation (PUE) attack. To prevent this attack, there should be a way to authenticate primary users' spectrum usage.

We propose a method that allows primary users to add a cryptographic link signature to its signal so the spectrum usage by primary users can be authenticated. This signature is added to the signal in a transparent way, such that the receivers (who do not care about the signature) still function as usual, while the cognitive radio receivers can retrieve the signature from the signal. We describe two schemes to add a signature, one using modulation, and the other using coding. We have analyzed the performance of both schemes.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication, Physical security, Unauthorized access*

General Terms

Security

Keywords

Cognitive Radio Networks, Primary User Emulation Attack, Physical-layer Authentication

1. INTRODUCTION

Recently, there has been a growing interest in cognitive radio. In general, cognitive radio refers to a wireless device that can change its transmission or reception parameters to achieve efficient communication [13]. One of the promising applications of cognitive radio is to enable the current fixed spectrum channels assigned by Federal Communications Commission (FCC) to be utilized by new users.

In a recent study, FCC has found that most of the radio frequency spectrum was inefficiently utilized [22]: some spectrum is overloaded, while other spectrum is rarely used. This is because spectrum is statically assigned by FCC; only the FCC-designated owners can use the spectrum assigned to them. With cognitive radios, FCC is considering allowing unlicensed users to utilize licensed bands provided it would not cause any interference (by avoiding transmission whenever licensed user's presence is sensed). This is a new paradigm for wireless communication. The licensed user is called *primary user*, and the unlicensed user is called *secondary user*.

Threat in cognitive radio network: To avoid interfering with primary users, secondary users should conduct primary user detection, i.e. to detect whether a primary user is using its spectrum or not. There are two main approaches for primary user detection: *energy detection* and *feature detection* [14]. In energy detection, secondary users use energy strength to identify a primary user's signal, whereas in feature detection, secondary users find some specific features of a signal, and use these features to identify a primary user. Examples of features include pilot, synchronization word, and cyclostationarity [10, 19, 20].

Unfortunately, neither energy detection nor feature detection can produce a trustworthy result. If a malicious secondary user wants to gain an unfair use of the primary user's idle spectrum, it can emulate the primary user's behavior or energy strength when sending its own signals. This attack is called Primary User Emulation (PUE) attack. In cognitive radio networks, it is essential to be able to detect whether the legitimate primary user is using the spectrum or not even if the network is under PUE attacks.

Existing approaches: The problem is actually an authentication problem, i.e., when a receiver has detected sig-

*This work was supported in part by AFOSR Award No. FA9550-09-1-0224 and by a subcontract from ANDRO Computational Solutions under the AFOSR STTR program contract FA9550-10C0179.

WiSec'11, June14–17, 2011, Hamburg, Germany.
Copyright 2011 ACM 978-1-4503-0692-8/11/06 ...$10.00.

nals at a particular spectrum, how can the receiver be sure that the signal is indeed sent by the primary owner of the spectrum? In general, this is a solved problem, as we can simply ask the primary user to attach a digital signature in its signals. Unfortunately, we are facing three constraints.

First, in cognitive radio networks, it is impractical to conduct authentication at layers other than the physical layer. As we know, authentication can be done at various levels, including data-link layer (e.g. wireless access point), network layer (e.g. IPSec), transport layer (e.g. SSL), and application layer (e.g. SSH). Unfortunately, conducting authentication above the physical layer does not seem practical in the cognitive radio applications. The reason is the following. For two devices to be able to authenticate each other at Layer N, these devices normally need to have a common component at Layer N. For example, to use IPSec for authentication purpose, both devices must run the IP and IPSec protocols. In the Internet, most of the computers are running IP, so this is not an issue. Unfortunately, in the wireless world, devices are so diversified, many of them differ significantly above the physical layer. For example, a cognitive radio receiver may be able to receive signals from TV stations, process them at the physical layer, but it may lack the component to understand the data in the signals. Therefore, if the authentication depends on the correct understanding of the data (done at upper layers), the cognitive radio receiver will be unable to authenticate the primary user.

Second, the authentication scheme should be transparent to the existing receivers, i.e., after the authentication information is added to signals, the existing devices should still be able to work as usual, although they may not be able to authenticate the signals. This requirement is extremely important. For example, when a TV station adds some authentication information to its signals for cognitive radio receivers, if all the existing television sets (which are not cognitive radios) need to be recalled and modified, no TV station will add the authentication information. Therefore, it is essential for the authentication scheme at the physical layer to be transparent to the existing devices.

Third, to further complicate the problem, FCC made another rule, stating that "no modification to the incumbent system (i.e., primary user) should be required to accommodate opportunistic use of the spectrum by secondary users" [5].

Several solutions have been proposed to solve the authentication problem. One approach is to use location detection [4]. Namely, secondary users detect whether a signal is indeed from the primary user's location. Such an approach either needs to use sophisticated antenna or collaboration of multiple nodes; both are expensive.

Recently, Liu et al proposed an interesting solution to solve the authentication problem while complying to the FCC regulation [16]. In this scheme, a helper node is placed close to a primary user. The idea is to put necessary mechanisms on the helper node, which then conveys the authentication information to the secondary users. While this scheme complies with the FCC restriction, the cost is quite significant. For example, establishing such a node is not easy or cheap. To cover the same area as its primary user, the helper node needs to use the same level of energy when transmitting (a typical Digital TV tower covers over 50 miles radius). Although the helper node does not need to transmit at all the time, its overall energy usage is significant, as it needs to wake up quite frequently to serve the newly joined secondary users, who can join at any arbitrary time.

Our approach. We have a different take on the FCC regulation. The main reason for this regulation is the cost induced on the primary users. If the cost is too high, primary users will be reluctant to participate. The cost can be measured from three aspects: the existing receiver's equipment update costs, the primary user's equipment update costs, and the operation cost. Since the existing receivers are often in a large quantity for most primary users, any solution that requires the update on the existing receivers will never be adopted. Therefore, the main concern of the FCC regulation boils down to the cost on the primary users, including the operating costs and an one-time equipment update cost.

The ideal solution is to follow the FCC rule while paying no cost or minimal cost. Such a solution has yet to be discovered and will be definitely an interesting direction to pursue. The state-of-art solution proposed by Liu et al. [16] complies with the FCC rule, but pays a heavy cost on both operation and equipment. If the FCC's real concern is the cost, a solution with lower cost should become promising even if the FCC rule is not followed. Unlike the other FCC rules, there is no negative impact on the community if this rule is violated. We strongly believe that if we can demonstrate significant benefit, FCC may consider lifting this rule. After all, by allowing cognitive radios to use primary users' spectrum, FCC is actually lifting a pre-existing rule. Moreover, FCC rules only apply to the U.S., other counties may not have such a rule.

For this technical paper, we have no intention of arguing whether FCC should lift that rule (the decision is more political than scientific). Our objective is to show the research community as well as to FCC the followings using scientific evidences (instead of political arguments): (1) the FCC regulation can be lifted without affecting the existing receivers, and (2) without this FCC regulation, the cost of authenticating primary users can be significantly reduced by using the ideas proposed in the paper.

Our problem, constraints, and challenges. Summarizing the above discussion, we formulate the following objective for this paper:

> Our objective is to develop low-cost physical-layer schemes for authenticating primary users' spectrum usage. The schemes should be transparent to the existing receivers.

We have developed two techniques, one based on the QPSK modulation, and the other based on error-correcting codes. We have conducted comprehensive analysis to demonstrate the effectiveness of our schemes.

2. PHYSICAL-LAYER SPECTRUM USAGE AUTHENTICATION

The problem we are addressing in this paper seems like a broadcast authentication problem [18]; however, compared to the existing broadcast authentication problem, our problem has the following unique properties. First, the goal of our problem is to authenticate the *spectrum usage*, i.e., to verify whether the primary user is indeed using a specific spectrum; we do not need to verify whether the *contents* sent by the primary user are authentic or not. On the other hand, verifying the authenticity of the contents is exactly the objective of broadcast authentication.

Second, there are two types of receivers in our problem. One is the cognitive radio receivers (called *CR Receivers* in short). They need to be able to authenticate the spectrum usage from the signals sent by the primary users. Because this type of receivers is not the dedicated "listener" of a primary user, they may not have the capabilities (e.g. circuit or software) to understand the primary user beyond the physical layer. Therefore, the authentication has to be done at the physical layer. The solutions to the broadcast authentication problem do not have such a constraint, so they are mostly developed in the upper layers.

The other type of users is the existing receivers. They are, in many cases, not cognitive radios. They have no interest to verify whether a spectrum is actually used by a primary user or not. We call this type of receivers the *non-CR* receivers. Because the physical-layer functionalities are usually built into the hardware for these receivers, it is difficult and costly to modify these receivers. Therefore, the solution to our problem should not require any change to the non-CR receivers. The traditional broadcast authentication problem does not have such a constraint.

Based on the unique features of our problem, we decompose our problems into two independent problems:

- **Problem 1 (Tag Generation):** What kind of information should be used for authenticating spectrum usage? Namely, how can primary users generate authentication tags (we refer to the authentication information as *authentication tag* in this paper), so CR receivers can use the tag to verify whether a spectrum is currently being used by its legitimate owner or not?
- **Problem 2 (Tag Transmission):** How can primary users transmit authentication tags, so the tags can be retrieved by CR receivers from the signal at the physical layer, while the tags do not interfere with non-CR receivers' functionalities. In other words, authentication tags should be *transparent* to non-CR receivers.

2.1 Authentication Tag Generation

Problem 1 can be solved using one-way hash chains. We describe the solution in the following.

Preparation: The primary user generates the following one-way hash chain:

$$h_n \rightarrow h_{n-1} \rightarrow \ldots \rightarrow h_1 \rightarrow h_0,$$
$$\text{where } h_i = hash(h_{i+1}).$$

The end of the hash chain, h_0, should be published beforehand so all CR receivers can get it. For example, h_0 can be published on a web site [1]. Each number on the hash chain is only valid in a specific time window. We use $[t_{i-1}, t_i]$ to represent the effective time window for the hash value h_i. The hash chain has to be used reversely, i.e., h_1 will be used first (during $[t_0, t_1]$), then h_2 (during $[t_1, t_2]$), and so on. Because of the way how the one-way hash chain is generated, disclosing h_i does not lead to the disclosure of h_j for $j > i$.

Authentication Tag. For the primary user, between time t_{i-1} and t_i, the authentication tag is simply h_i, i.e., the primary user simply embeds h_i to its signals (how to embed the value to signals will be discussed later). An example is shown in Figure 1. In this example, during $[t_1, t_2]$, h_2 is embedded in the signals, and sent out repeatedly. The repetition is necessary because CR receivers may tune in to this spectrum at any arbitrary moment.

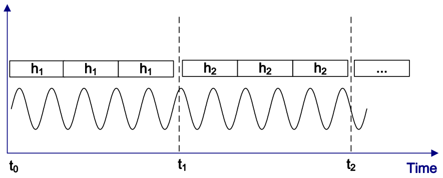


Figure 1: Authentication Tags

Tag Verification. Once a CR receiver receives the signals from a particular spectrum, it retrieves the authentication tag h_i from the signals; then using the current time and the spectrum owner's h_0 value[2], the receiver can verify the validity of h_i. It should be noted that only loose time synchronization is needed in this scheme.

Replaying Attacks. At first sight, this solution seems problematic, because a malicious receiver can replay the authentication tag in its own signals once the valid h_i is disclosed by the primary user. This is actually not a big problem. Recall the goal of h_i is to prove to the receivers that the primary user is using the current spectrum at time t, where $t \in [t_{i-1}, t_i]$. If the primary user is still using the spectrum, then replaying h_i has no negative effect, because whatever the attack says is still the truth. However, if the primary user stops using the spectrum, then the attacker's replaying of h_i will have a negative effect, as the attacker can successfully fool the receivers. Nevertheless, such attack is only effective within the $[t_{i-1}, t_i]$ time window. When the time window expires, attackers need h_{i+1} to continue to fool the receivers. However, if the primary user is not using the spectrum, h_{i+1} will not be sent out.

Therefore, the maximum advantage that a malicious user can take is the length of a time window, i.e., they can unfairly use the rest of a time window if the primary user stops. As we will show later, the actual time window can be set to quite to a small value, thus minimizing the impact of such an attack. For example, for Digital TV broadcasting, our results show that the time to transmit one 128-bit tag is only 3.2 ms. If we allow 5 seconds of error (in both directions) in clock synchronization, an 11-second window will be sufficient.

When the time window gets smaller, the hash chain will be longer for a fixed period of time. If a primary user stops using a spectrum for a long period of time, receivers need to do many hash functions to verify an authentication tag when the primary user becomes active. This is not a major problem, because hash operations are quite efficient. However, if such a cost is a concern for CR receivers, we can use multiple hash chains. For example, we can use a different hash chain each day, so every day, we start with a new h_0 value. All these h_0 values can be distributed together.

[1] The web site should use `https` to protect the authenticity of h_0.

[2] Spectrum assignments are public knowledge, and can be obtained beforehand.

2.2 Authentication Tag Transmission

After the authentication tag is generated, we need to find a way to embed the tag in the signals. Generally speaking, this should not be very difficult; however, we are bounded by two constraints. First, tags have to be added at the physical layer. This is because most primary users differ quite significantly above the physical layer. For a CR receiver to authenticate different primary users at upper layers require the CR receiver to be equipped with the corresponding hardware or software to understand the protocols at upper layers. This is expensive and unrealistic. Our second constraint is the transparency requirement. In order not to disrupting service to existing receivers, the added authentication tag should not require any modification for the non-CR receivers.

The question is whether any information can be added to the physical layer without changing the physical-layer behaviors of the non-CR receivers. The answer is yes. Actually, in wireless communication, extra data are constantly "added" to the existing signals; these data are called *noise*. During the transmission of wireless signals, noise are always present in the received signals. Therefore, the physical-layer logic has to be designed to tolerate noise up to certain degree.

Our idea is to treat the authentication tag as noise ("man-made" noise), and then intentionally add the noise to the signals. If we can keep the noise level low enough, the physical-layer logic of the non-CR receiver will naturally filter out the noise. On the other hand, if we can keep the noise level above certain threshold, the CR receivers will be able to retrieve the "man-made" noise (i.e. tags) from the signals.

Our main challenge is to find a place in the physical layer, where the "man-made" noise can be added. There are two main components in the physical layer: coding and modulation. For coding, there are two different types: one is source coding, the goal of which is to increase the efficiency of transmission; the other is channel coding, the goal of which is to improve the reliability of transmission. We focus on the channel coding component. For modulation, its goal is to transform a message signal (e.g. a digital bit stream or an analog audio signal) into suitable format that can be physically transmitted. Authentication tags can be added to both coding and modulation components (see Figure 2). We will discuss them separately in the following two sections.

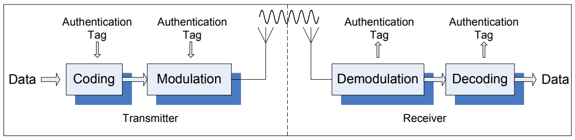


Figure 2: Process

3. ADDING TAGS TO MODULATION

In this section, we describe how authentication tags can be transparently added to modulation schemes at the physical layer. There are many modulation schemes, and the way how tags are added will be quite different, although the essence is the same, i.e., tags are added as noise. In this paper, to present a concrete method, we have chosen a popular modulation scheme, called QPSK (Quadri-Phase-Shift Keying). We describe how tags can be added to this particular scheme. The technique can be easily generalized to broader classes of modulations, such as PSK (Phase Shift Keying), of which QPSK is a special case.

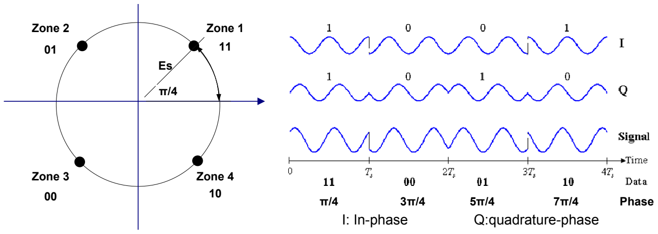


Figure 3: QPSK Modulation

3.1 QPSK Background

QPSK [12] is one of the digital modulation techniques used for transmission of digitally represented data. This modulation scheme uses phases in the transmitted wave to carry information. In QPSK, four phases are used: $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ and $\frac{7\pi}{4}$. These four phases can carry two bits of information. Therefore, in the input data, which is a sequence of binary data stream, each two bits are treated as a pair. There are four combination of pairs: 00, 01, 10 and 11, and each of these unique combinations is called a *dibit*. Each dibit is mapped to one phase. For example, if we choose the Gray encoded set of dibits, 11, 01, 00 and 10 are mapped to phases $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ and $\frac{7\pi}{4}$, respectively. This mapping is called a *dibit-phase mapping*.

With this dibit-phase mapping, we can modulate the signal using the following signal modulation equation:

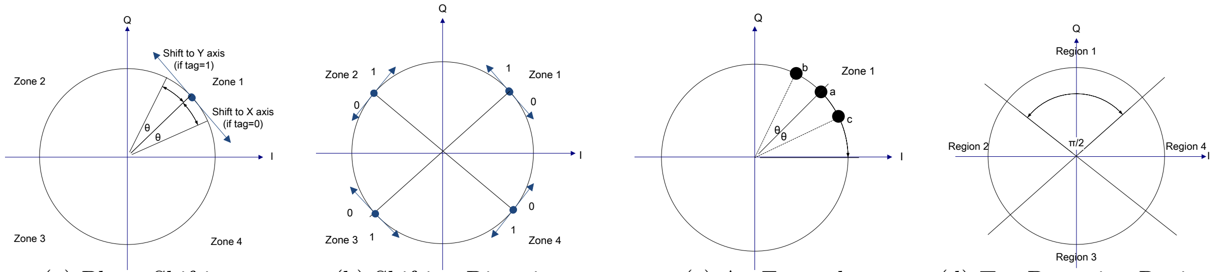
$$S_i(t) = \sqrt{\frac{2E_s}{T}}\cos(2\pi f_c t + (2i-1)\frac{\pi}{4}) \qquad i = 1, 2, 3, 4.$$

A QPSK signal can actually be represented by a two-dimensional signal constellation, which is a common representation of signal in digital modulation schemes. QPSK constellation is considered as a diagram for dibits-phase mapping. The diagram is divided into four zones by the x-axis and y-axis. There are four message points in the constellation diagram, each falling into one zone. These points represent messages 11, 01, 00 and 10; their corresponding phases are $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, and $\frac{7\pi}{4}$, respectively. These points' signal strength is the same (equal to E_s), so they are placed on the same circle. Figure 3 depicts the constellation diagram.

In QPSK modulation, the digital data stream is transformed to the resulting QPSK signal and then the transmitter would send it out. When a receiver gets the signal, it generates a QPSK constellation from the received signal. Then, depending on which zone a signal falls into, the receiver can determine the dibit information in the signal.

3.2 QPSK Tagging Scheme

Because of noise, when receivers receive a signal, and convert the signal into the QPSK constellation diagram, the message points may not fall exactly on their original positions, i.e. at $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ or $\frac{7\pi}{4}$. They may scatter around the original positions, i.e., noise may have perturbed the positions. That is why in the demodulation, receivers will use the zone, instead of the positions, to get the data carried by the signals. As long as the positions are not perturbed



(a) Phase Shifting (b) Shifting Directions (c) An Example (d) Tag Detection Regions

Figure 4: The QPSK Tagging Scheme

too much, the data carried by the signals can still be correctly retrieved. In other words, QPSK can tolerate noise to certain degree.

We can take advantage of this noise tolerance. We can treat our tag as a man-made noise, and intentionally perturb the message points during modulation based on the tag information. When the receiver gets the signal, it can get the tag information based on how the message points are perturbed. We discuss this process in more details in the following modulation and demodulation parts.

Adding Tags in Modulation. The basic idea to add man-made noise (i.e. tags) to QPSK is to perturb the position of the phase in the constellation diagram, just like what natural noise does to the phase positions. There are many different ways to perturb the positions, we will only describe a simple perturbation method in the following (our analysis will be based on this method):

- When the tag is 1, we shift the phase by θ degree towards the y-axis, where $0 < \theta < \frac{\pi}{4}$. The final position stays on the circle to maintain the same signal energy.
- When the tag is 0, we shift the phase by θ degree towards the x-axis.

Figures 4(a) and 4(b) illustrate the tagging scheme. Figures 4(c) shows an example of adding tag information to signals in Zone 1. In this example, the data is 11. If the tag is 1, the final phase position will be at point *b*; otherwise, it will be at point *c*.

Retrieving Tags in Demodulation. When demodulating, we need to retrieve both data and tag. As we have already discussed, data can be retrieved based on which zone in the constellation diagram the signal falls into. As for the tag, we divide the constellation into four *regions*. Different from zones, the boundary of regions are the two cross lines in Figure 4(d). Based on zones and regions, receivers can detect both data and tag.

- Data detection: this is the same as the original QPSK scheme, i.e., data are detected based on the zones. Keeping data detection scheme the same is essential for the transparency for the non-CR receivers.
- Tag detection: if data fall into Region 1 and Region 3, the carried tag bit is 1; otherwise, it is 0.

Obviously, due to the natural noise and our man-made noise, there will be errors in both data and tag detections. We will present the detailed analysis results in Section 5.

4. ADDING TAGS TO CODING

In this section, we describe how authentication tags can be transparently added to the coding module at the physical layer. In particular, we focus on the coding module that tries to enhance the error tolerance of communication systems. A common scheme used in this module is the Error Correcting Code (ECC).

4.1 ECC Background

Error correcting codes provide a mechanism for improving the error performance of communication systems. This is achieved by adding redundancy. For example, in order to transform a message of k symbols, we can use an encoding scheme to add redundant information in a particular way to map this message into an n-symbol codeword ($n > k$), such that the codeword can tolerate up to t corrupted symbols. Such a code is referred to as a (n, k) block code. We call t the error correction capability of this code i.e., it can correct up to t symbol errors per n-symbol codeword.

One of the most common class of ECC code is the Reed-Solomon code. For example, The digital TV broadcast uses the $(207, 187)$ Reed-Solomon code [1], i.e. each 187-symbol input block is turned into a 207-symbol block, where each symbol consists of 8 bits. In Reed-Solomon code, error correction capability t equals to $\frac{n-k}{2}$. Therefore this code can tolerate up to 10 corrupted symbols in the communication ($t = \frac{207-187}{2} = 10$ in the $(207, 187)$ code).

4.2 ECC Tagging scheme

Our main idea of adding authentication tags in the ECC module is to take advantage of the error correction capability t of the code. Basically, to embed a tag, we intentionally corrupt symbols at certain particular positions in the transmitted codeword. As long as the total number of errors (our "injected" errors plus the errors naturally incurred) in each codeword are still less than t, the error correction code module at the receiver side will be able to recover all the symbols correctly in the codeword. Therefore, non-CR receivers can receive the signals as usual, i.e., the tags are transparent to this type of receivers. In our scheme, we embed the tag information in certain position of the codeword such that the receiver can extract the tag information before the receiver decodes the codeword. The explanation of our scheme is given in below.

Consider a communication system that uses an (n, k) linear block code to improve its error performance, where each symbol consists of M bits. Let $(\mathbf{c_1}, \ldots, \mathbf{c_L})$ be a sequence of codewords that need to be transmitted. Our goal is to

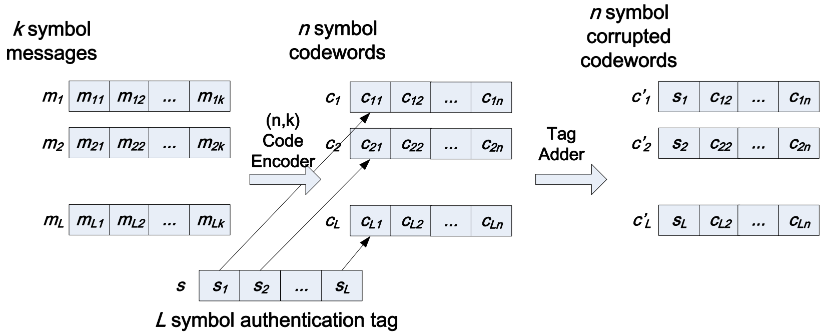


Figure 5: The ECC Tagging Scheme

embed an authentication tag in this sequence. Let $\mathbf{s}$ be our tag, it is divided into L symbols, each with M bits, i.e., $\mathbf{s} = (s_1, \ldots, s_L)$. To encode the authentication tag $\mathbf{s}$ in the codeword sequence $(\mathbf{c_1}, \ldots, \mathbf{c_L})$, we replace the first symbol in $\mathbf{c_i}$ with s_i (for $i = 1, 2, \ldots, L$). The process is illustrated in Figure 5.

In the above description, we only corrupt one symbol in each codeword. We can corrupt more than one symbols in each codeword to embed more tag information. There is a tradeoff. By adding authentication tag in our described manner, we are effectively reducing the error correcting capability of the code. Namely, if the transmitter intends to insert q tag symbols (i.e., corrupting q symbols) in a codeword with error correction capability of t symbols, then we are effectively reducing its actual capability to $(t-q)$. Finding a right balance between q and t is very important.

Moreover, it should be noted that our tag can also be corrupted due to the errors incurred in the communication. Therefore, receivers may be unable to get the tag with 100 percent accuracy. We will analyze this error probability in the next section.

5. ANALYSIS

The objective of this section is to understand the performance of the QPSK and ECC tagging schemes. In particular, we would like to study how the tagging schemes affect data transmissions (i.e., the data error rate), and how well receivers can recover the tag (i.e., the tag error rate). The following parameters are relevant to our scheme:

- The length L of each tag: Each tag is basically a hash value. In our analysis, we assume that MD5 is used, i.e., each tag has 128 bits. Similar analysis can be done for other hash functions.

- Signal-to-noise ratio (SNR), E_b/N_0: It is defined as the power ratio between a signal (meaningful information) and the background noise (unwanted signal), and measured by dB.

In this section, we first analyze the following error rates. These analytical results will be used as the basis for our more comprehensive analysis in Sections 6 and 7.

- Signal symbol error rate P_s: This probability indicates how accurately receivers can retrieve the data sent in the signal. P_s is defined upon *symbol*. In QPSK, a symbol consists of two bits. In the ECC tagging scheme, a symbol is made of M bit. The value of M depends on what error correcting code is used.

- Bit error rate for tag detection P_t: This probability indicates how accurately cognitive radio receivers can retrieve each bit of the tag information.

5.1 Analysis of the QPSK Approach

In QPSK, the offset to the signal constellations affects the signal symbol error probability. To be more specific, when the shifting angle offset becomes larger, the error performance for the data becomes worse, but the error on detecting the tags will decrease. Therefore, it is a tradeoff between the signal symbol error rate and the tag bit error rate. We would like to analyze and understand this tradeoff.

Let us analyze the situation in Figure 4(a) (because of symmetry, it is sufficient to analyze only one zone): Assume that the modulated signal $S(t, \theta)$ consists of two parts: signal symbol (dibit) 11 and authentication tag 1. Since the tag is 1, the shifting direction is to the Y-axis. Let the shifting angle be $\theta \in (0, \pi/4)$. After the transmission, the received signal $\bar{S}(t, \theta)$ is the following: (We assume the Additive White Gaussian Noise (AWGN) model in our analysis)

$$\bar{S}(t,\theta) = S(t,\theta) + W(t) = \sqrt{\frac{2E_s}{T_s}}\cos(2\pi f_c t + \frac{\pi}{4} + \theta) + W(t),$$

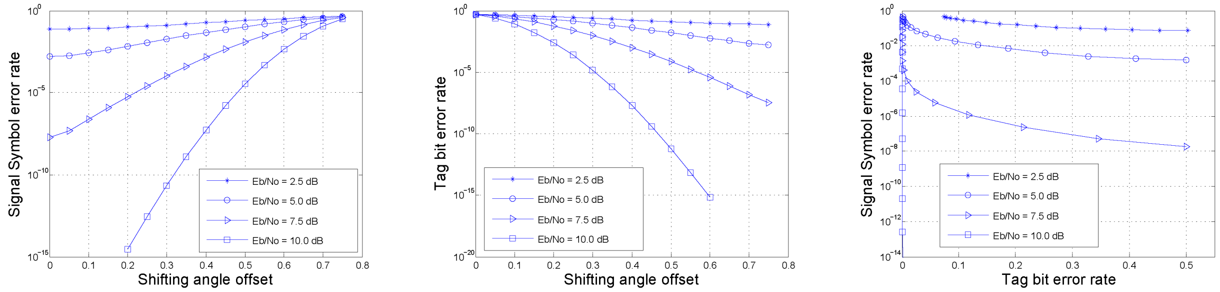
where $W(t)$ is the sample function of a white Gaussian noise process of zero mean and power spectral density $N_0/2$; A detailed analysis of $S(t, \theta)$ is in Appendix $A.1$.

According to its received signal $\bar{S}(t, \theta)$, the observation vector s of a coherent QPSK receiver has two elements:

$$\begin{aligned} x_1 &= \sqrt{\frac{E_s}{2}}(\cos\theta - \sin\theta) + w_1, \\ x_2 &= \sqrt{\frac{E_s}{2}}(\cos\theta + \sin\theta) + w_2. \end{aligned}$$

Thus, x_1 and x_2 are sample values of independent Gaussian random variables with mean values equal to $\sqrt{E_s/2}(\cos\theta - \sin\theta)$ and $\sqrt{E_s/2}(\cos\theta + \sin\theta)$, respectively, and with a common variance equal to $N_0/2$. These two elements together decide the positions of the received signal $\bar{S}(t, \theta)$ in the QPSK constellation.

Based on our decision rules in the QPSK tagging scheme, we can calculate the signal symbol error rate P_s and the tag bit error rate P_t:



(a) Data: Symbol Error Rate

(b) Tag: Bit Error Rate

(c) Tradeoff Curve

Figure 6: Performance of the QPSK Tagging Scheme

Theorem 1. *Let the angle offset be θ. Let E_b/N_0 be the signal-to-noise ratio (SNR), where $E_b = E_s/2$. The signal symbol (2 bits per symbol in QPSK) error rate P_s is given in the following formula:*

$$\begin{aligned} P_s \quad &\simeq \quad \frac{1}{2}\operatorname{erfc}\Big(\sqrt{\frac{E_b}{N_0}}(\cos\theta - \sin\theta)\Big) + \\ &\qquad \frac{1}{2}\operatorname{erfc}\Big(\sqrt{\frac{E_b}{N_0}}(\cos\theta + \sin\theta)\Big), \\ \textit{where}, \quad \operatorname{erfc}(x) \quad &= \quad \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}\,dt \end{aligned}$$

To prove the theorem, we simply need to compute the following probability (a proof is given in Appendix A.2):

$$P_s = 1 - Pr(\text{s falls inside Zone 1}).$$

We plot the results of P_s in Figure 6(a). From the curves, we can see that when the shifting angle increases, P_s also increases, i.e., the performance of data detection gets worse. If we fix the shifting angle, and compare the three curves, we can see that the larger the SNR is, the lower the P_s is; therefore increasing SNR can reduce data errors.

The next theorem gives the bit error rate for tags.

Theorem 2. *Let the angle offset be θ. Let E_b/N_0 be the signal-to-noise ratio (SNR), where $E_b = E_s/2$. The bit error rate P_t for tag is the following:*

$$\begin{aligned} P_t \quad &\simeq \quad \frac{1}{2}\operatorname{erfc}\Big(\sqrt{\frac{E_b}{N_0}}(\cos\theta)\Big) + \frac{1}{2}\operatorname{erfc}\Big(\sqrt{\frac{E_b}{N_0}}(\sin\theta)\Big), \\ \textit{where} \quad &\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}\,dt. \end{aligned}$$

To prove the theorem, we simply need to compute the following probability (a proof is given in Appendix A.3):

$$P_t = 1 - Pr(\text{s falls inside Region 1 or Region 3}).$$

We plot the results of P_t in Figure 6(b). Each curve indicates that as the angle offset increases, P_t decreases, i.e., the performance of QPSK tagging scheme gets better. It seems that Figure 6(a) and Figure 6(b) are quite similar in shapes; that is because the functions used to calculate both signal symbol error rate and tag error rare are essentially the same for QPSK, only with different parameters.

To understand the relationship between data errors and tag errors, we plot them together in Figure 6(c). We can clearly see the tradeoff between these two errors. We can also see that increasing SNR can achieve a better performance for our tagging scheme. When SNR equals 10 dB, the curve is a straight line. That's because the tag bit error rate is very low at that SNR.

5.2 Analysis of the ECC Approach

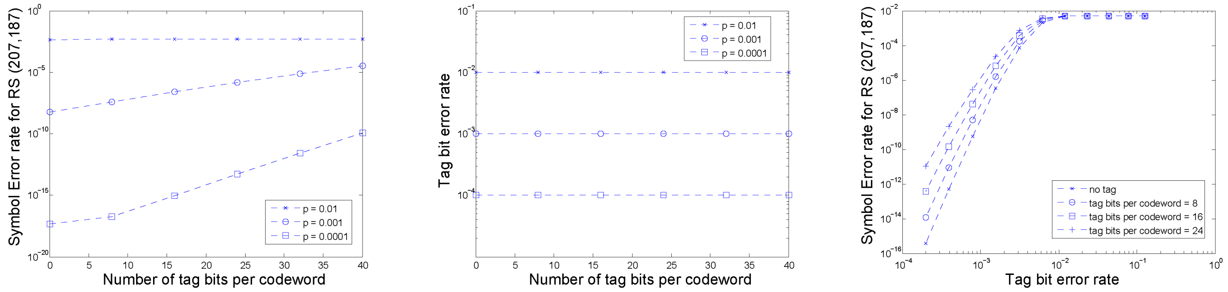
To perform the analysis for the ECC tagging scheme, we assume a binary symmetric channel (BSC) with channel error rate p. A Binary Symmetric Channel is one where the input symbol to the channel is 1 bit and the probability of receiving erroneous symbol is p. This means, if the input to the channel is 1(0), we receive 0(1) with probability p. The purpose of using a BSC is to hide the underlying modulation schemes, interleaving and other communication blocks, so as to make the analysis independent to any of them. To compute the bounds for the error probabilities we assume that hard decision [3] is made by the receiver.

The three important metrics that we are going to show are signal symbol error rate P_s after decoding codeword, codeword error rate P_{cw}, and tag bit error rate P_t. The codeword error rate P_{cw} for a given code is the probability that the transmitted codeword and the decoded codeword are not the same. Similarly, the symbol error rate P_s is the probability that the transmitted symbol and the received symbol are not the same. Tag bit error rate P_t is the probability that the received tag bit is in error.

We would like to analyze our tagging scheme using a specific type of linear block code. We choose the Reed-Solomon (RS) codes, which are a very important class of error correcting codes; it finds application in a large number of digital communication systems. Let an (n, k) RS code encode a k-symbol input message into a n-symbol output message. Let each symbol be M bits wide. This (n, k) RS code can correct up to $t = \frac{n-k}{2}$ symbol errors per codeword [2].

Now, if we corrupt q ($q < t$) symbols in each codeword, the codeword error rate P_{cw} and symbol error rate P_s are

[3] We say a receiver makes a *hard decision* when each received analog symbol at modulation layer is quantized and decoded to the respective bits independently of other received symbols.



(a) Data: Symbol Error Rate

(b) Tag: Bit Error Rate

(c) Tradeoff Curve

Figure 7: Performance of the ECC Tagging Scheme

bounded by the followings [2]:

$$
\begin{aligned}
P_{cw} &\leq \sum_{i=(t-q)+1}^{n} \binom{n}{i} p_s^i (1-p_s)^{n-i}, \\
P_s &\leq \frac{1}{n} \sum_{i=(t-q)+1}^{n} i \binom{n}{i} p_s^i (1-p_s)^{n-i}, \\
& \text{where, } p_s = 1-(1-p)^M.
\end{aligned}
$$

By the nature of our scheme, corrupting q symbols in a codeword is equivalent to replacing that many symbols in the codeword with q symbols of the authentication tag, with each symbol consisting of M bits.

Regarding the tag bit error P_t, since the tag bits in a received codeword are checked before being handed to the error correction code decoder, the bit error rate for the tag is the same as the channel bit error p of the BSC channel under consideration. Therefore,

$$P_t = p.$$

To plot the performance of this scheme we use the (207,187) Reed Solomon code. Each symbol in a codeword of this code is 8 bits wide. A plot of symbol error rate P_s against the number of tag bits per codeword is shown in Figure 7(a). It can be seen that as we embed more bits in a codeword, the error performance of the code decreases. The behavior of tag bit error rate P_t against the number of tag bits per codeword is shown in Figure 7(b). Because the way we receive tag bits is independent of the decoding of codeword, we can see that the tag bit error remains constant even if we add more bits per codeword. The fact that tag bit error rate is equal to the channel error rate and it is constant results in its positive correlation with the symbol error rate as shown in Figure 7(c).

6. REDUCING TAG ERROR RATE

The authentication tag is formed using cryptographic hash functions, so, even a single bit error in the tag will make the tag invalid. In order to transmit the tag reliably, its error probability needs to be brought down to a fairly small value. However, the analysis in the last section show that to achieve a reasonably low error rate on data, the bit error rate of tag is not so low. Therefore, for a 128-bit MD5 hash, having at least one bit of error in the tag is quite possible. We use P_e^{tag} to represent the error rate for the entire L-bit tag, i.e. P_e^{tag} is the probability of having at least one bit error in the L-bit tag. This is different from the bit error rate P_t discussed in the previous section.

We need to keep P_e^{tag} fairly low (e.g., below 10^{-10}), even if the bit error rate P_t is not low. There are two approaches to do so. One is the repetition approach, i.e., we repeat each tag for many times. This way, even if some bits of a tag are corrupted during the transmission, receivers can still re-construct the correct tag with high probability by combining all the copies of the tag together (e.g. using majority voting for each bit). Another approach is to use error correcting code on the tag. Error correcting code can be used to correct the error bit in the tags and bring down tag error rate. Because our tag is quite short (e.g. 128 bits for MD5), we can afford to use a long code to keep P_e^{tag} significantly small. We will discuss and analyze this approach in this section.

Let the L-bit authentication tag be encoded using an (n^{tag}, k^{tag}) linear block code, which can correct up to t errors. Let each symbol in the code be 1 bit wide. Let P_t be the tag bit error probability, which is already given in Section 5 for both QPSK and ECC tagging schemes. In our analysis, if we assume a binary symmetric channel for tag transmissions, P_t is essentially the channel error rate. For most of the linear block codes it is very difficult to find the exact codeword error rate or bit error rate. Therefore, they are upper bounded by the following inequality [2]:

$$P_{cw}^{tag} \leq \sum_{i=t^{tag}+1}^{n^{tag}} \binom{n^{tag}}{i} P_t^i (1-P_t)^{n-i}.$$

$$P_b^{tag} \leq \frac{1}{n^{tag}} \sum_{i=t^{tag}+1}^{n^{tag}} i \binom{n^{tag}}{i} P_t^i (1-P_t)^{n-i}.$$

Since the tag is encoded by ECC code, we are interested in finding the whole L-bit tag error rate. It is given by the following theorem.

Theorem 3. *Let L be the length of the tag. Let (n^{tag}, k^{tag}) be the linear block code that we use to encode this tag. Let D be the desired upper bound for the codeword error rate P_{cw}^{tag}.*

The probability (P_e^{tag}) that there is at least one bit of error in the tag can be upper bounded by the following inequality:

$$P_e^{tag} \leq 1 - (1-D)^{\frac{L}{k^{tag}}}.$$

The proof is given in Appendix A.4. It should be noted that D is selected by users, and it decides n^{tag}, i.e. which RS code we need to choose. The smaller the D is, the larger the n^{tag} will be.

7. EVALUATION

Now, we are going to answer the question that are essential to primary and secondary users, i.e., given a bound on P_e^{tag}, what are the impacts of our tagging schemes on them? For primary users, they worry about how much error performance (mostly reliability) may be affected by the proposed scheme, or how much power they need to increase to keep the same level of performance. For secondary users, they worry about how long it takes them to verify a tag. Because secondary users need to monitor a quite large spectrum range (i.e., many primary users), they cannot afford to spend too long on one primary user: the shorter, the better. Formally speaking, primary users concern about their transmission powers (i.e., the signal to noise ratio) and data error rate (i.e., signal symbol error rate in our analysis), while secondary users concern about the time for receiving a complete tag.

To facilitate our evaluation, we define the tag to data ratio W as the ratio of tag rate over data rate, which is used for calculating how long it takes to transmit a single authentication tag. We also define the ratio of k to n for any (n, k) linear block code, which is called code rate. Therefore, for the (n^{tag}, k^{tag}) code used to encode our tag, its code rate is $R_c^{tag} = \frac{k^{tag}}{n^{tag}}$.

The following two theorems give the tag to data ratio W for QPSK and ECC schemes respectively.

THEOREM 4. *In the QPSK tagging scheme, let R_c^{tag} be the code rate for the encoded tag, if encoding tags by a (n^{tag}, k^{tag}) linear block code. The tag to data ratio W for QPSK scheme is given by the following (the proof is given in Appendix A.5):*

$$W = \frac{R_c^{tag}}{2}.$$

THEOREM 5. *In the ECC tagging scheme, let R_c^{data} be the code rate for the encoded data. Let R_c^{tag} be the code rate for the encoded tag. If q is the number of encoded tag bits we embed in each n bit data codeword, then the tag to data ratio W is given by the following (the proof is given in Appendix A.6):*

$$W = \frac{qR_c^{tag}}{nR_c^{data}}.$$

This tag to data ratio W decides how long it takes a cognitive radio receiver to get a complete tag. Since the verification of a tag is quite fast (computing a few one-way hash functions), we will not include the tag verification time in our evaluation. Based on the tag to data rate W, the time T^{tag} required to transmit one L-bit authentication tag can be computed using $T^{tag} = \frac{L}{W \cdot r}$, where r is the data rate (bits per second) in a communication system.

Now, we are ready to find out two important relationships among signal to noise ratio (SNR), data error rate (i.e. signal symbol error rate in our analysis), and tag to data ratio. The first is the relationship between the data error rate and the tag to data ratio W, if the primary users decide to keep the same transmission power (i.e. SNR) as that without the authentication tag. The second relationship is the one between the transmission power and the tag to data ratio W, if the primary users decide to keep the data error rate the same as that without the tag.

For both QPSK and ECC tagging schemes, the essential issue is to choose a proper (n^{tag}, k^{tag}) code to keep the tag error rate below a threshold. Namely, when SNR and data error rate are fixed, the length of tag is decided by the tag's error correcting code that we select to achieve our threshold goal. The bigger the code rate is, the bigger the tag to data ratio, so the smaller the tag transmission time T^{tag} becomes.

Based on the theorems derived in this paper, we are able to find a group of suitable (n^{tag}, k^{tag}) codes to bring the error rate of a 128-bit tag down to the threshold ε. Assuming ε is 10^{-10} (i.e., $P_e^{tag} < 10^{-10}$), we are able to plot Figures 8(a) and 8(b) for the QPSK tagging scheme, and Figures 8(c) and 8(d) for the ECC tagging scheme.

Since it is difficult to test all the (n^{tag}, k^{tag}) codes, the code we use is not guaranteed to be the optimal one. These four figures only show the basic relationship between SNR, data error rate, and tag to data ratio; they are not precisely the boundary or optimal solution. Solution using the optimal codes can achieve a better result. Finding the optimal solution is one of the directions in our future research.

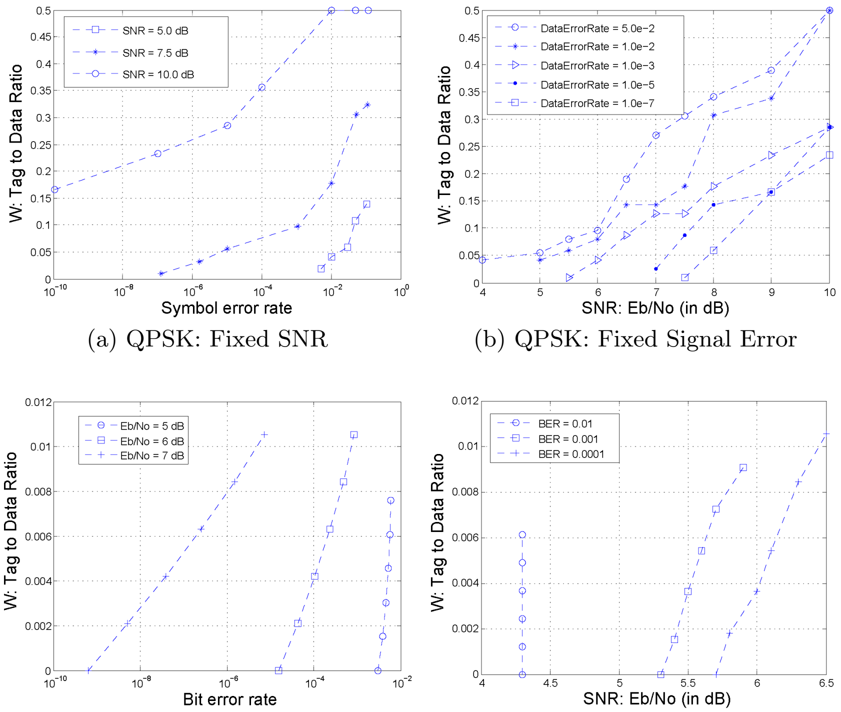
QPSK tagging scheme evaluation. Figure 8(a) shows that W increases as the data error rate increases (Symbol Error Rate in QPSK), when the transmission power is kept the same (assuming that the noise power does not change, increasing SNR means increasing signal power). This means that if the primary users want to achieve a higher tag to data ratio with the same power, they have to sacrifice data reliability, that is, increasing data error rate.

Figure 8(b) shows, in each curve, if the primary users want to keep the same data error rate, W will increase if the SNR increases. Therefore, if we augment the signal power, the tag to data ratio is going to increase, and therefore less time is required to send a tag.

On both Figures 8(a) and 8(b), the largest value of tag to data ratio W is 0.5. That is because the best case (i.e., tags do not need to be encoded with error correcting codes) in QPSK is to embed one bit of tag for each two data bits.

An Example. There are many applications of QPSK in reality. Consider the Digital Video Broadcasting Satellite, to which we can apply the QPSK Tagging scheme. Suppose we would like to keep our 128-bit tag error rate below 10^{-10} and keep the symbol error rate for a receiver below 10^{-5}. We also assume SNR = 8 dB and tag bit error rate P_t is 5×10^{-3}. In this situation, as showed in Figure 8(b), tag to data ratio W is 0.15. Since the data rate for the QPSK is 39Mbps due to DVB-S [17], the time required to transmit one authentication tag is $\frac{128}{0.15 \times 39M} = 2.18 \times 10^{-2}$ms.

ECC tagging scheme evaluation. As for ECC tagging scheme, to have a common basis for comparison between the tagging schemes, we assume that QPSK modulation is used to transmit the bits in the ECC tagging scheme too. Assuming the channel to be memoryless i.e., the data errors are independent of each other, the channel error rate, p, for



(a) QPSK: Fixed SNR

(b) QPSK: Fixed Signal Error

(c) ECC: Fixed SNR

(d) ECC: Fixed Signal Error

Figure 8: Comprehensive Analysis Of QPSK and ECC Tagging Schemes

a AWGN channel is given by [2], $p = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right)$, where E_b/N_0 = Signal to Noise ratio (SNR). From this expression we can see that for a constant SNR, the channel error rate p is constant.

A look at Figure 7(a) tells us that by corrupting bits in a codeword we effectively reduce the error performance of the code. Therefore, to maintain the same error performance at the receiver we need to increase SNR; if SNR is fixed, the error performance will degrade. Figure 8(c) shows the relationship between the tag to data ratio W and the data error rate at a constant SNR for the RS (207, 187) code. Figure 8(d) shows the relationship between the tag to data ratio W against SNR at a constant data error (for the same RS code).

An Example. Error correcting codes are used in Digital TV broadcasting. Suppose the channel error rate at a DTV receiver is 10^{-3}. Since it uses RS (207,187) code with symbol width of 8 bits, each codeword contains $207 \times 8 = 1656$ bits. If we encode the 128-bit tag using (127, 50) BCH code (One kind of Error-correcting Code), which has symbol width of 1 bit, the length of encoded tag will be $127 \times 128/50 = 326$ bits. Afterwards, every 8 bits of encoded tag are embedded into each 1656-bit codeword, the tag to data ratio can be calculated to be, $W = 2.044 \times 10^{-3}$. Since, the data rate for the regular terrestrial DTV transmission is 19.39 Mbps [1], the time required to transmit one authentication tag is $\frac{128}{2.044 \times 10^{-3} \times 19.39 \times 10^{6}} = 3.2\text{ms}$.

8. RELATED WORK

Numerous work has been conducted to derive the signal features that are unique to a transmitter or primary user, so these features can be used as signatures to identify a particular transmitter [3, 8, 11] or detect legitimate primary users [10, 14, 19–21, 24]. Although this approach has been successful in certain scenarios, recently, it was pointed out by Danev et al. that the features are not completely trustworthy, and most features can be spoofed [7].

A recent attempt concerns adding secure signatures to transmitter signals for authentication. Wang et al. [23] proposed a scheme to add a low-power signal as the identity in television broadcast. Yu et al. proposes a physical-layer authentication scheme [25]. Their idea is to superimpose a tag signal with the original signal in order to prove its legitimacy to receivers. However, the authentication scheme described in the paper is based on a secret key; this is not practical for broadcasting authentication. Moreover, the tag in the paper is generated from the signal and the secret key; this makes the authentication quite sensitive to errors: if there is an error in the data (which is quite common in the physical layer), it will be hard to verify the authentication tag. Furthermore, the way how tags are added to signals in our work is different from that in [25]. As for the solution proposed by Liu et al. [16], we have already discussed it in the introduction section.

Other related work includes water marking [6,9,15], which also discuss how to embed tags inside signals. Their objective is different from ours; their goal is to use the tag for the copyright protection purpose.

9. CONCLUSION

In this paper, we present a method to solve the primary emulation attack in cognitive radio networks. In our scheme, primary users use a one-way hash chain to authenticate its legitimate use of spectrum. To ensure that the existing non-CR receivers can properly receive signals, the authentication tag must be transparent to them. Thus, we present two schemes, one conducting tagging in QPSK modulation, the other in the error-correcting coding. We have analyzed the performance of two schemes, and our evaluation results indicate that the scheme is quite practical for cognitive radio.

10. REFERENCES

[1] ATSC Digital Television Standard Part 2: RF/Transmission System Characteristics (A/53, Part 2:2007. `www.atsc.org`.

[2] J. G. Proakis and M. Salehi, *Digital Communications.* New York, McGraw-Hill, 2007, p. 192, 434, 472-474.

[3] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Proceedings of the ACM MobiCom*, September 14–19 2008.

[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*, (1):25–37, 2008.

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

[6] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as commnications with side information. *Proceedings of the IEEE*, (7):1127–1141, July 1999.

[7] B. Danev, H. Luecken, S. Čapkun, and K. E. Defrawy. Attacks on physical-layer identification. In *Proceedings of the 3rd ACM Conference on Wireless Network Security*, 2010.

[8] B. Danev and S. Čapkun. Transient-based identification of wireless sensor nodes. In *Proceedings of the 8th IEEE/ACM Information Processing in Sensor Networks*, pages 25–36. IEEE/ACM, 2009.

[9] C. Fei, D. Kundur, and R. H. Kwong. Analysis and design of secure watermark-based authenticaiton systems. *IEEE Transactions on Information Forensics and Security*, (1):43–55, March 2006.

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

[11] J. Hall, M. Barbeau, and E. Kranakis. Detecting rogue devices in bluetooth networks using radio frequency fingerprinting. In *Proceedings of Communications and Computer Networks*, pages 108–113, October 4–6 2006.

[12] S. Haykin. Book: Digital communications.

[13] J. Mitola III and G. Q. Maguire, Jr. Cognitive radio: making software radios more personal. *IEEE Personal Communications Magazine*, (4):13–18, August 1999.

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

[15] J. E. Kleider, S. Gifford, S. Chuprun, and B. Fette. Radio frequency watermarking for ofdm wireless networks. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2004.

[16] Y. Liu, P. Ning, and H. Dai. Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures. In *Proceedings of 2010 IEEE Symposium on Security and Privacy*, May 2010.

[17] Introduction of DVB-S. Website: http://www.complextoreal.com/tdvbs.htm.

[18] A. Perrig and J. D. Tygar. *Secure Broadcast Communication: in Wired and Wireless Networks.* Kluwer Academic Publisher, 2003.

[19] Y. Qi, T. Peng, W. Wang, and R. Qian. Cyclostationarity-based spectrum sensing for wideband cognitive radio. In *Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, pages 107–111, 2009.

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

[21] 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*, 2006.

[22] G. Staple and K. Werbach. The end of spectrum scarcity. *IEEE Spectrum*, March 2004.

[23] X. Wang, Y.Wu, and B. Caron. Transmitter identification using embedded pseudo random sequences. *IEEE Transactions on Broadcasting*, (3):244–252, September 2004.

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

[25] P. L. Yu, J. S. Baras, and B. M. Sadler. Physical-layer authentication. *IEEE Transactions on Information Forensics and Security*, (1), March 2008.

APPENDIX

A. PROOFS

A.1 Detaild Analysis of QPSK

Without authentication tags, the original modulated signal is the following:

$$S(t) = \sqrt{\frac{2E_s}{T_s}} \cos(2\pi f_c t + \pi/4).$$

After adding the tag, the new modulated signal becomes

$$S(t,\theta) = \sqrt{\frac{2E_s}{T_s}} \cos(2\pi f_c t + \pi/4 + \theta)$$

$$= \sqrt{\frac{E_s}{2}}(\cos\theta - \sin\theta)\phi_1(t) - \sqrt{\frac{E_s}{2}}(\cos\theta + \sin\theta)\phi_2(t),$$

where, $\phi_1(t)$ is the in-phase component basis function, and $\phi_2(t)$ is the quadrature-phase component basis function:

$$\phi_1(t) = \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t), \quad \phi_2(t) = \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t).$$

A.2 Proof of Theorem 1

PROOF. In this QPSK Tagging Scheme, signal symbol error probability is the following:

$$\begin{aligned} P_t &= Pr(\text{s falls inside any regions except Region 1}) \\ &= 1 - Pr(\text{ s falls inside Region 1}) \\ &= 1 - Pr(\text{ } x_1 \text{ falls inside Region 1 }) * \\ &\quad Pr(\text{ } x_2 \text{ falls inside Region 1 }), \end{aligned}$$

where x_1 and x_2 are sample values of independent Gaussian random variables with mean values equal to $\sqrt{E_s/2}(\cos\theta - \sin\theta)$ and $\sqrt{E_s/2}(\cos\theta + \sin\theta)$, respectively, and with a common variance equal to $N_0/2$. Moreover, x_1 and x_2 are independent of each other.

Therefore,

$$P_t = 1 - \int_0^\infty \frac{1}{\sqrt{\pi N_0}} exp[-\frac{(x_1 - \sqrt{\frac{E_s}{2}}(\cos\theta - \sin\theta))^2}{N_0}]\, dx_1 * \\ \int_0^\infty \frac{1}{\sqrt{\pi N_0}} exp[-\frac{(x_2 - \sqrt{\frac{E_s}{2}}(\cos\theta + \sin\theta))^2}{N_0}]\, dx_2$$

Let

$$\frac{x_1 - \sqrt{\frac{E_s}{2}}(\cos\theta - \sin\theta))}{\sqrt{N_0}} = z_1$$

$$\frac{x_1 - \sqrt{\frac{E_s}{2}}(\cos\theta + \sin\theta))}{\sqrt{N_0}} = z_2$$

Then changing the variables from x_1 to z_1, and x_2 to z_2, we can have

$$\begin{aligned} P_t = 1 - \{1 - \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta - \sin\theta))\} * \\ \{1 - \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta + \sin\theta))\} \\ = \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta - \sin\theta)) \\ + \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta + \sin\theta)) \\ -\frac{1}{4}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta - \sin\theta)) * \operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta + \sin\theta)). \end{aligned}$$

In the region where $(E_s/N_0) \gg 1$, we may ignore the second term on the right side of *Eq.* 1, so approximate the formula for average signal symbol error probability as

$$\begin{aligned} P_s &\simeq \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta - \sin\theta)) + \\ &\quad \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_s}{2N_0}}(\cos\theta + \sin\theta)), \\ where, \quad \operatorname{erfc} &= \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}\, dt \end{aligned}$$

In QPSK, two bits per symbol, which means the signal energy per bit is half of signal energy per symbol, that is,

$$E_b = \frac{1}{2}E_s$$

Thus, we may write:

$$\begin{aligned} P_s &\simeq \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_b}{N0}}(\cos\theta - \sin\theta)) + \\ &\quad \frac{1}{2}\operatorname{erfc}(\sqrt{\frac{E_b}{N0}}(\cos\theta + \sin\theta)) \end{aligned}$$

□

A.3 Proof of Theorem 2

The proof is similar to the proof of Theorem 1, so we omit it here.

A.4 Proof of Theorem 3

PROOF.

$$\begin{aligned} P_e^{tag} &= Pr\{\text{at least 1 error in tag}\} \\ &= 1 - Pr\{\text{no error in any tag codeword}\} \\ &= 1 - (1 - P_{cw}^{tag})^{\frac{L}{k^{tag}}} \\ &\leq 1 - (1 - D)^{\frac{L}{k^{tag}}} \end{aligned}$$

□

A.5 Proof of Theorem 4

PROOF. Suppose, we encode the tag using a (n^{tag}, k^{tag}) code to bring down the tag error probability below a certain threshold. Then, the code rate for the tag is

$$R_c^{tag} = \frac{k^{tag}}{n^{tag}}.$$

In case of QPSK, we superimpose 1 bit of encoded tag on every symbol, which represent 2 bits. So, the ratio of tag rate to data rate is:

$$W = \frac{R_c^{tag}}{2}.$$

□

A.6 Proof of Theorem 5

PROOF. Let, the communication system use an (n, k) code. Then the code rate in this case is

$$R_c^{data} = \frac{k}{n}.$$

Suppose, we encode the tag using a (n^{tag}, k^{tag}) code to bring down the tag error probability below a certain threshold. Then, the code rate for the tag is

$$R_c^{tag} = \frac{k^{tag}}{n^{tag}}.$$

If, we embed q bits of encoded tag in n bits of a codeword, then the number of tag bits per n bit codeword is qR_c^{tag}

Then, the ratio of tag rate to data rate is given by

$$W = \frac{qR_c^{tag}}{k} = \frac{qR_c^{tag}}{nR_c^{data}}.$$

□