Exploiting Spectrum Sensing Data for Key Management

Exploiting Spectrum Sensing Data for Key Management

Ahmed Badawy[a], Tarek Elfouly[b], Carla-Fabiana Chiasserini[a], Tamer Khattab[c], Daniele Trinchero[a]

[a] *Politecnico di Torino - DET (ahmed.badawy, chiasserini, daniele.trinchero@polito.it)*
[b] *Qatar University, Computer Science and Engineering Dept. (tarekfouly@qu.edu.qa)*
[c] *Qatar University, Electrical Engineering Dept. (tkhattab@qu.edu.qa)*

Abstract

In cognitive radio networks, secondary users (SUs) communicate on unused spectrum slots in the frequency bands assigned to primary users (PUs). Like any other wireless communication system, cognitive radio networks are exposed to physical layer attacks. In particular, we focus on two common attacks, namely, spectrum sensing data falsification and eavesdropping. Such attacks can be counteracted by using symmetric key algorithms, which however require a complex key management scheme. In this paper we propose a novel algorithm that significantly reduces the complexity of the management of symmetric link keys by leveraging spectrum sensing data that is available to all nodes. In our algorithm, it is assumed that a primary secret key is pre-distributed to the legitimate SUs, which is needed every number of detection cycles. With the aid of the information provided in the primary key, our algorithm manipulates the collected samples so that a segment of the estimated sensing statistic at the two legitimate SUs can be used as a seed to generate a common symmetric link key. The link key is then employed to encrypt the transmitted data. Our algorithm exhibits very good performance in terms of bit mismatch rate (BMR) between two link keys generated at the two legitimate SUs. In addition, our solution is robust against the difference in the received signal to noise ratio between two legitimate SUs thus making it suitable for practical scenarios. Furthermore, our algorithm exploits the decision statistic that SUs use for spectrum sensing, hence, it does require neither extra processing nor extra time, allowing the SUs to quickly and securely tab into empty spectrum slots.

Keywords: Spectrum sensing, Key management, Cognitive radio network, Eavesdropping, Spectrum sensing data falsification.

1. Introduction

In cognitive radio networks, a secondary user (SU) accesses the spectrum whenever the spectrum owner, named primary user (PU), is not transmitting, or both PU and SU can share the spectrum under the PU's defined terms of usage. Consequently, reliable spectrum sensing (SS) and decision algorithms at SUs are paramount for the detection of spectrum holes. One of the main signal detection techniques is the General Likelihood Ratio Test (GLRT) [1], which is used when detection is performed in the presence of some unknown parameters [2]. Once the adopted SS technique detects an empty spectrum slot, SUs can use it to communicate.

On the other hand, securing the communication between legitimate SUs is a challenging issue due to the fact that numerous attacks can be launched against cognitive radio networks. Comprehensive studies on this aspect [3, 4, 5] show that two of the major physical layer attacks against cognitive radio networks are spectrum sensing data falsification (SSDF) and eavesdropping. SSDF is performed on a collaborative sensing setup [6]: an attacker sends false spectrum sensing data to other SUs, in case of distributed sensing decision, or to the fusion center [7], resulting in a wrong spectrum access decision. Eavesdropping attackers instead are adversaries or unauthorized users that listen to the communication between legitimate users.

Conventional techniques to combat SSDF leverage a two-level defense mechanism [8]. The first level authenticates all the collected spectrum sensing results, while the second decides which spectrum sensing result is legitimate. Depending on whether a fusion center is available or the system is fully distributed, schemes such as the sequential probability ratio test (SPRT) [8], or reputation-based schemes can be exploited [8]. Techniques designed to counteract SSDF, however, require a long processing time for the two stages to occur. Moreover, either a large number of SUs or many successful iterations are needed to achieve a good reputation. Clearly, long processing time might lead to higher probability of missing the opportunity of exploiting empty spectrum slots for SUs. In addition, authentication techniques such as the approach in [9], where cyclo-stationary detection is used to classify and authenticate signals, adds to the complexity and limitations of the system, while failing to prevent a scenario where a malicious node mimics the SU's signal properties.

Alternatively, physical layer security techniques exploit the randomness inherent to communication channels, which are common to the two trusted parties and unknown to a potential eavesdropper, so as to generate secret keys [10, 11, 12]. Although these algorithms were not developed for cognitive radio network applications, they can be utilized by the SUs. However, physical-layer solutions, such as channel estimation based on one or two level defence mechanisms, involve exchange of multiple beacon signals as well as synchronization between legitimate SUs thus requiring a long time to generate the link key and, hence an inefficient usage of the spectrum.

To counteract eavesdropping, a power allocation approach is proposed in [13] to increase the secrecy level between authenticated SUs. Alternatively, conventional wireless security, which relies on cryptographic techniques and application-layer protocols, can be adopted [14]. Fundamentals of key management protocols are presented in [15, 16, 17]. One drawback of these techniques however is that a complex key management scheme is required in the case of symmetric ciphers, while high computational complexity is needed in the case of asymmetric ciphers. In particular, in the case of symmetric ciphers, the continuous exchange of encryption keys poses a serious threat to the secrecy of the whole communication session. Minimizing the security risk that stems from key exchange mechanisms is the main reason for key reuse (i.e., using the same key for multiple packet encryptions), which introduces another secrecy weakness allowing an eavesdropper to have more chances to guess the encryption key.

In this paper, we propose a novel technique to counteract the two aforementioned cognitive radio attacks and at the same time solve the above shortcomings of symmetric key management schemes such that it suits the peculiarity of cognitive radio networks. Our key management technique comprises a primary key distribution and a link key generation algorithm. We exploit the spectrum sensing data available to all nodes (legitimate and malicious) and the secret primary key to generate *link* keys. We assume that the secret *primary* key is pre-distributed to legitimate SUs, which then use the information provided therein to manipulate the samples collected via SS. By applying a decision algorithm, which is in our case the GLRT, on the manipulated samples, a segment of the estimated decision statistic at the legitimate SUs is used as the seed to generate the secret *link* key, which is

employed to encrypt the transmitted data [1]. Note that the SS process is not affected by such operations. In other words, the manipulated collected samples are used for the purpose of sensing the spectrum and generating a link key. While the sensing statistics may be available also at the malicious node, thanks to the secret primary key, the link key is only available at the legitimate SUs and unknown to the malicious node. A new link key is generated for each new detection cycle. Also, our algorithm does require neither any extra extensive processing nor time to secure the link between SUs, both in the case of distributed collaborative detection and in the case where a fusion center is available since the centralized processor is usually an SU. In addition, our algorithm does not require any extensive beacon exchange as in the case of physical layer techniques [18]. To the best of the authors' knowledge, the idea of exploiting spectrum sensing data to extract a secret link key has never been studied in the literature.

The rest of the paper is organized as follows. Section 2 describes the system model. The adopted decision algorithm, which is based on GLRT, is introduced in Section 3. Section 4 presents our link key generation algorithm. Results are then shown in Section 5, while Section 6 concludes the paper.

2. System Model

Consider a radio cognitive network where the SUs sense the spectrum so as to detect empty spectrum slots that they can exploit for communication, i.e., the spectrum is already occupied by the PU's signal and the objective is to determine the gaps in the PUs communications. While communicating, SUs periodically sense the spectrum in order to be able to detect the entrance of a PU and, in case, retreat from using the spectrum slot. The time intervals corresponding to the two operations are referred to as Phase I and Phase II, respectively, and they are depicted in Figure 1. A detection cycle is defined as the time period comprising the two phases. Note that the length of the detection cycle and of the two phases therein is not constant. Indeed, PUs exploit their assigned spectrum as desired and, therefore, the length of each phase may differ from one detection cycle to the next.

We assume that every N samples, each SU makes its own decision about the frequency slot status (empty/occupied) and sends it to the other SUs, in

[1] Note that link key is often derived from a primary key (see for example the Bluetooth specifications and temporal key integrity protocol in 802.11).

case of distributed collaborative sensing, or to the SU acting as fusion center. The first objective is to ensure that the decisions collected from all SUs, which will be used to produce the overall decision on the presence or absence of PUs, are validated and false samples generated by malicious nodes are discarded. To this end, decisions from legitimate SUs are encrypted with a *link* key only known to them. A decision maker can then easily decrypt the data and filter out information injected by malicious nodes. Similarly, in the presence of an empty spectrum slot, legitimate SUs encrypt their communication through a *link* key, so as to avoid eavesdropping by a malicious user.

A link key is generated by SUs at every detection cycle. In order to do that, we assume that an authorized network entity distributes a secret primary key to legitimate SUs prior to the spectrum sensing operation, using any of the conventional cryptographic schemes presented in [19]. The primary key includes information that is essential to the algorithm we devise to generate the link key. Note that the primary key is also delivered to any legitimate SU that joins the network later on. A new primary key is instead distributed whenever its effect on the link key generation diffuses with time (e.g., every number of detection cycles as discussed in Sections 4 and 5), and whenever a legitimate SU leaves the network. The latter is necessary to secure the network against the scenario when a legitimate SU later becomes a malicious node.

Finally, our adversary model assumes that a malicious node can listen to the spectrum used by the PU and can use the same SS technique used by the legitimate SUs. In other words, the malicious node has access to the spectrum sensing data. The malicious node's intention is to launch an SSDF attack by transmitting false spectrum sensing data to the other SUs, or to the SU operating as fusion center. It can move freely within the field and can visit any of the locations where either the PU or the SUs were or will be. In the case of eavesdropping, the malicious node is assumed to be a passive adversary.

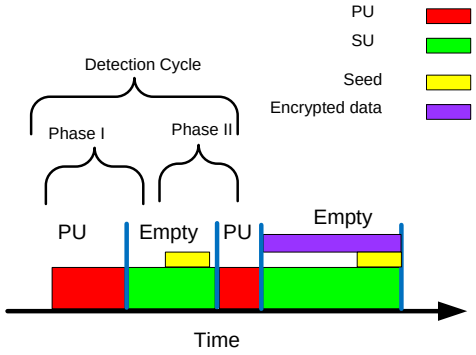

Figure 1: Spectrum sensing and link key generation during each detection cycle.

3. Spectrum Sensing and Decision Algorithm

A SU, listening to a specific frequency band, collects samples $y[i]$, $i = 1, \ldots, N$. If the spectrum slot is empty (hypothesis H_0), $y[i] = w[i]$, where $w[i]$ is the additive white gaussian noise (AWGN) with variance σ_w^2. σ_w^2 is receiver dependant and can be estimated ahead of time. If instead the PU is transmitting (hypothesis H_1), $y[i] = x[i] + w[i]$, where $x[i] = hs[i]$ is the product of the channel gain h and the PU's signal $s[i]$. $x[i]$ is assumed to be Gaussian distributed with zero mean and variance σ_x^2. The value of σ_x^2 depends on the channel gain and the power of the PU signal. Thus, in the presence of the PU's signal, $y[i]$ follows a Gaussian distribution $\mathcal{N}(0, \sigma_w^2 + \sigma_x^2)$ [2, 20, 21], which we denote by F_1. Instead, in the case of an empty frequency band, $y[i]$ follows $\mathcal{N}(0, \sigma_w^2)$, which we denote by F_0.

In the remainder of this section, we first summarize the GLRT algorithm presented in [2], which is used for the detection of the entrance of a PU's signal. This algorithm is used during Phase II of the detection cycle. Then, we present our GLRT algorithm for the detection of empty spectrum slot, which is used during Phase I of the detection cycle.

3.1. Detection of the entrance of the PU's signal

The authors in [2] presented a GLRT-based algorithm for the detection of the entrance of the PU's signal. In the presence of an empty spectrum slot, the samples collected by the SUs follow distribution F_0 with density function f_0, and

$$y[i] = w[i], \qquad \text{for} \qquad i = 1, \cdots, k-1. \tag{1}$$

where k is the time instant at which the change of the frequency slot status is detected. As the PU enters the frequency band, the distribution changes to F_1 with density f_1, and

$$y[i] = x[i] + w[i], \qquad \text{for} \qquad i = k, \cdots, N \tag{2}$$

where recall that N is the number of samples corresponding to the periodicity with which SUs make their SS decisions. The scenario we are interested in is when σ_w^2 is known and σ_x^2 is in the range $[\sigma_{\text{S}}^2, \sigma_{\text{M}}^2]$. With $l_1(y)$ being the log-likelihood ratio, note that:

$$\sum_{i=\hat{k}+1}^{N} l_1(y[i]) = \ln \left\{ \prod_{i=\hat{k}+1}^{N} \frac{f_{1,\sigma_x^2}(y[i])}{f_0(y[i])} \right\}, \tag{3}$$

where $\hat{k}$ is the sampling time at which the SU estimates that the PU has entered the spectrum slot, and after which $l_1(y)$ shows a consistent positive drift. We denote the decision statistic for the detection of the entrance of the PU's signal by B_N. The decision statistic is estimated through [2]:

$$\begin{aligned} B_N &= \max_{\hat{k} \leq N} \sup_{\sigma_x^2} \ln \left\{ \prod_{i=\hat{k}+1}^{N} \frac{f_{1,\sigma_x^2}(y[i])}{f_0(y[i])} \right\} \\ &= \max_{\hat{k} \leq N} \sum_{i=\hat{k}+1}^{N} \left\{ \frac{1}{2} \ln \left\{ \frac{\sigma_w^2}{\sigma_x^2 + \sigma_w^2} \right\} + \frac{\sigma_x^2 y^2[i]}{2(\sigma_x^2 + \sigma_w^2)\sigma_w^2} \right\}, \end{aligned} \tag{4}$$

where f_{1,σ_x^2} is the probability density function of the received signal with the actual variance of the PUs signal being σ_x^2. B_N is compared against a threshold, λ_B, to decide on the presence or absence of the PU's signal.

3.2. Detection of empty spectrum slots

We develop the GLRT algorithm to detect the transmission opportunities, i.e. empty spectrum slots, rather than detecting the entrance of the PU's signal to be used during Phase I of our system model. Again, at first the samples collected by the SU follow distribution F_1 with density function f_1 (hypothesis H_1), and

$$y[i] = x[i] + w[i], \qquad \text{for} \qquad i = 1, \cdots, k-1, \tag{5}$$

As the PU leaves the frequency band, the distribution changes to F_0 with density f_0 (hypothesis H_0) and $\exists k \in [1, N]$

$$y[i] = w[i], \qquad \text{for} \qquad i = k, \cdots, N. \tag{6}$$

With $l_2(y)$ being the log-likelihood ratio in this case, note that:

$$\begin{aligned}\sum_{i=\hat{k}+1}^{N} l_2(y[i]) &= \ln\left\{\prod_{i=\hat{k}+1}^{N} \frac{f_0(y[i])}{f_{1,\sigma_x^2}(y[i])}\right\} \\ &= \sum_{i=\hat{k}+1}^{N}\left\{\frac{1}{2}\ln\left\{\frac{\sigma_w^2+\sigma_x^2}{\sigma_w^2}\right\} - \frac{\sigma_x^2 y^2[i]}{2(\sigma_x^2+\sigma_w^2)\sigma_w^2}\right\}. \end{aligned} \tag{7}$$

Let:

$$f(\sigma_x^2) = (N-\hat{k})\frac{1}{2}\ln\left\{\frac{\sigma_x^2+\sigma_w^2}{\sigma_w^2}\right\} - \frac{\sigma_x^2 \widehat{y}}{2(\sigma_x^2+\sigma_w^2)\sigma_w^2}. \tag{8}$$

Since σ_x^2 is unknown, we find its estimate $\sigma_x^{2^*}$ by solving (8) for the value that maximizes it, which results in:

$$\sigma_x^{2^*} = \begin{cases} \sigma_{\text{M}x}^2, & (N-\hat{k}) \leq \frac{\widehat{y}}{\sigma_{\text{M}x}^2+\sigma_w^2}, \\ \frac{\widehat{y}}{N-\hat{k}} - \sigma_w^2, & \frac{\widehat{y}}{\sigma_{\text{M}x}^2+\sigma_w^2} \leq (N-\hat{k}) \leq \frac{\widehat{y}}{\sigma_{\text{S}x}^2+\sigma_w^2}, \\ \sigma_{\text{S}x}^2, & (N-\hat{k}) \geq \frac{\widehat{y}}{\sigma_{\text{S}x}^2+\sigma_w^2}, \end{cases} \tag{9}$$

where $\widehat{y} = \sum_{i=\hat{k}+1}^{N} y^2[i]$. Consequently, for a preset N, an iterative $\hat{k}$ and $\sigma_x^{2^*}$ estimated through (9), the decision statistic, denote by E_N, is given by:

$$\begin{aligned} E_N &= \max_{\hat{k}\leq N} \sup_{\sigma_x^2} \ln\left\{\prod_{i=\hat{k}+1}^{N} \frac{f_0(y[i])}{f_{1,\sigma_x^2}(y[i])}\right\} \\ &= \max_{\hat{k}\leq N} \sum_{i=\hat{k}+1}^{N}\left\{\frac{1}{2}\ln\left\{\frac{\sigma_w^2+\sigma_x^{2^*}}{\sigma_w^2}\right\} - \frac{\sigma_x^{2^*} y^2[i]}{2(\sigma_x^{2^*}+\sigma_w^2)\sigma_w^2}\right\}. \end{aligned} \tag{10}$$

The decision statistic E_N is again compared to a threshold λ_E to decide on the presence or absence of the PU's signal according to: $E_N \underset{H_1}{\overset{H_0}{\gtrless}} \lambda_E$. The

threshold, $\lambda_B = -\ln\{a/b\}$, is set based on the average delay for false alarm $\overline{T_0} \geq 1/a$ where a is a design parameter and b is given by

$$b = 3\ln\left\{a^{-1}\left(1 + \frac{1}{D(f_0||f_{1,\sigma^2_{\mathrm{S}x}})}\right)^2\right\}, \tag{11}$$

with $D(f_0||f_{1,\sigma^2_{\mathrm{S}x}})$ being the Kullback-Leibler divergence of f_0 from f_1 estimated at $\sigma^2_{\mathrm{S}x}$. The Kullback-Leibler divergence of f_0 from f_1 is given by:

$$\begin{aligned} D\left(f_0||f_1\right) &= \mathbb{E}_{f_0}\left\{l_2\left(y[i]\right)\right\} \\ &= \int f_0(y)\ln\left\{\frac{f_0(y)}{f_1(y)}\right\}dy, \end{aligned} \tag{12}$$

where $\mathbb{E}$ denotes the expectation operator. Substituting f_0 and f_1 at $\sigma^2_{\mathrm{S}x}$ yields

$$D\left(f_0||f_{1,\sigma^2_{\mathrm{S}x}}\right) = \frac{1}{2}\ln\left\{\frac{\sigma^2_w + \sigma^2_{\mathrm{S}x}}{\sigma^2_w}\right\} - \frac{\sigma^2_{\mathrm{S}x}}{2(\sigma^2_{\mathrm{S}x} + \sigma^2_w)}. \tag{13}$$

4. Secret Key Generation Algorithm

In our proposed link key management algorithm, we will use the estimated decision statistic introduced before as a common seed for secret *link* key generation. We assume that all the legitimate SUs employ the same spectrum sensing algorithm, hence the decision statistic is already calculated at all the SUs. Below we first provide an outline of our algorithm and then we detail the steps on how to generate the secret link key from the decision statistic.

4.1. Algorithm Outline

The flow chart of our algorithm is presented in Figure 2. The algorithm is initialized at the first detection cycle, during Phase II. It is then repeated in Phase II of every cycle.

As mentioned, our technique consists of a primary key distribution and a link key generation algorithm. We assume that a primary secret key is pre-distributed to the legitimate nodes. Using the information provided in the primary pre-distributed key, our algorithm manipulates the samples collected by each legitimate SU during Phase II, i.e., when SUs are sampling white noise. Doing so, the estimated decision statistic at any two legitimate SUs

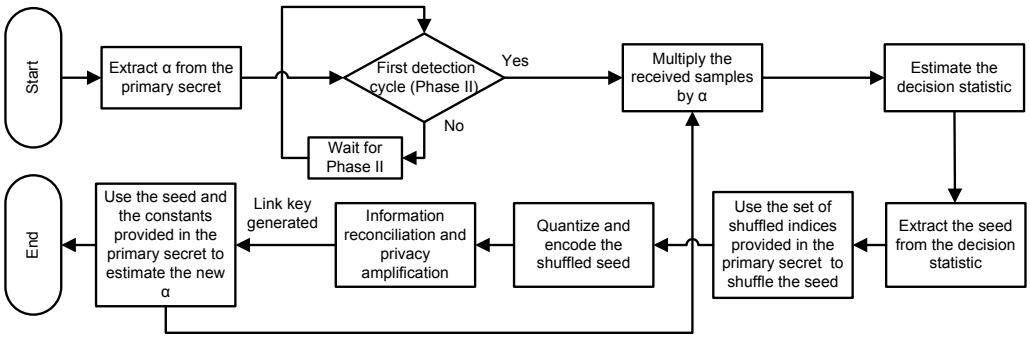

Figure 2: Flow chart of the proposed algorithm.

will be very similar[2], providing a common seed. This manipulation process is performed by applying at the legitimate SUs a mathematical operation on the collected samples, which can be as simple as a multiplication, or more complex such as a nonlinear function. For simplicity, here we assume a multiplication by a constant α, which, in the first cycle, coincides with one of the pieces of information included in the primary secret key and is then updated in the following cycles. Based on such samples, the decision statistic in (4) is computed.

Next, N_s samples are sequentially picked from the estimated decision statistic and used as seed, $S = [s_1, \ldots, s_{N_s}]$. S is shuffled, quantized using N_q bits and encoded to generate a serial bit stream. An information reconciliation and privacy amplification is applied to the generated serial bit stream in order to generate the final *link* key. S is then used to generate the new α for the following detection cycle through a pseudo-random number generator [22].

4.1.1. Counteracting SSDF and eavesdropping

The *link* key generated in one detection cycle is used in the two phases of the following detection cycle. The aims are twofold.

1. To counteract SSDF: during Phase I of the following detection cycle, the SS decision statistic estimated through (10) is encrypted with the generated *link* key and transmitted to the fusion center. The fusion

[2]The seed used for link key generation (explained later) is not exactly the same, but it is similar enough to act for link key generation. We will show that by plotting the bit mismatch rate between the generated links keys in Section 5.

center being one of the legitimate SU or another node having access to the spectrum has also generated the *link* key. Hence, it decrypts the transmitted SS decision statistic sent from legitimate nodes and easily filters out data sent from malicious nodes.

2. To counteract eavesdropping: once availability of an empty spectrum slot is declared, legitimate SU start communicating. Data is encrypted using the generated *link* key available at the legitimate SUs. An eavesdropper, which does not have the key, will not be able to decrypt the transmitted data.

4.2. Primary Key

As mentioned, the pre-distributed primary key is needed *only once* at the system set up, or after a number of detection cycles. This primary key is not the secret *link* key that will be used to encrypt the transmitted data. Rather, it contains some pieces of information that will be used in the process of generating the secret link key at two legitimate SUs. Specifically,

- the initial value of α;
- the set of the shuffled indices of the seed samples;
- the number of the compression function and universal hash function applied in the information reconciliation and privacy amplification step;
- the constants (β, γ and ρ) used in the process of generating the new value of α.

The number of quantization bits, N_q and the number of seed sample, N_S are fixed and given beforehand.

4.3. Seed Generation

In Phase II, each legitimate SU listens to the spectrum and collects AWGN samples before the entrance of the PU's signal. The SU first multiplies the samples by the initial value of α that is provided in the primary secret key, thus obtaining $y_\alpha[i] = \alpha y[i]$. Accordingly, distributions F_0 and F_1 change into $\mathcal{N}(0, \alpha^2(\sigma_w^2 + \sigma_x^2))$ and $\mathcal{N}(0, \alpha^2\sigma_w^2)$, respectively. We denote by B_α the decision statistic in (4) when y_α is used as input instead of y. Note that the likelihood ratio in (3) will also use y_α instead of y. Also, in Phase II, legitimate SUs may use either B or B_α for signal detection; clearly, in the latter case, the threshold used for SS should be adjusted accordingly.

Once B_α is available, the seed (S) is given by the N_s samples of B_α estimated before the entrance of the PU's signal. We will show that this seed does not depend on the signal-to-noise ratio (SNR) at the legitimate SUs but it mainly depends on α. This implies that, regardless of the received SNR, the generated seed can be considered common to all legitimate SUs making it suitable for secret *link* key generation.

4.4. Link Key Generation

The generation of the secret link key at legitimate users consists of the following four steps.

1) Once estimated the common seed, its indices are shuffled according to a sequence that is provided in the primary key. The main purpose of shuffling is to increase the level of randomness of the seed.

2) Next, the shuffled seed has to be converted into a bit stream that is suitable as link key. To quantize the seed samples, we use uniform quantization [23]. The number of quantization bits, N_q, determines the number of quantization levels, $L = 2^{N_q}$. The quantized decimal value is then converted into bits.

3) Although uniform quantization is easy to implement, increasing the quantization bit number dramatically degrades the performance of the algorithm since the Bit Mismatch Rate (BMR) between two communicating SUs increases. To solve this problem, we adopt the technique presented in [24]. There the authors proposed an encoding algorithm and applied it on a uniformly quantized reciprocal link signatures. On link signatures exhibiting a BMR up to 84.48%, their encoding scheme could reduce the BMR to almost 4% thus leading to an excellent improvement.

4) The final step towards the link key generation is information reconciliation, where the two legitimate SUs use a protocol, such as the one in [25], to minimize the BMR between bit streams generated at two different SUs. In this protocol, public communication over the channel must occur to correct the mismatched bits. Consequently, some of the information will be leaked to the eavesdropper. Therefore, information reconciliation is usually followed by data compression and privacy amplification where a universal compression function and a universal hash function is selected randomly from a saved set and applied to the bit streams at both the SUs [26]. The generated link key will then become shorter in length but higher in entropy. In our algorithm, the number of the compression function as well as the hash function is provided in the *primary* secret. It is worth noting that for the information

reconciliation step to be applied efficiently, the BMR after the encoding step should not exceed a certain value, namely, 15% [26]. After this step, the link key is generated and ready to be used to encrypt the transmitted data in the next cycle.

At last, SUs have to compute a new value of α to be used in the next detection cycle. To this end, the following operation is applied to the estimated seed:

$$S_{LGN} = \ln \mathbb{E}\left[S\right]. \tag{14}$$

S_{LGN} is the input to the Linear Congruential Generator (LGN) – a pseudo random number generator [22] requiring constants β, γ and ρ to compute the new value of α as:

$$\alpha = \left(\beta S_{LGN} + \gamma\right) mod(\rho). \tag{15}$$

where *mod* is the modulo operator. The constants β, γ and ρ are included in the primary secret.

We will use the root mean square error (RMSE) as the metric to evaluate the drift in computing α between two legitimate SUs, i.e., $RMSE = \sqrt{\mathbb{E}\left[\left(\alpha|_{SU_1} - \alpha|_{SU_2}\right)^2\right]}$.

5. Results

In this section, we present the simulation results for our proposed key management algorithm. We first present the simulation results for the estimated decision statistic in the two phases. We then show the effect of multiplying the received samples by α on the estimated decision statistic in Phase II. An example of shuffled seed is then illustrated. The effect of change in SNR and α on the BMR of the generated *link* key is then presented. We compare the bit mismatch and entropy rates of the key generated through our algorithm to conventional channel based algorithm. The simulated results for the BMR and entropy rates are presented after step (3) of our link key generation algorithm and before information reconciliation and privacy amplification steps. This is to show that the link key generated through our algorithm exhibits a good performance before these two standard steps. In addition, we depict how the value of α changes with different detection cycles.

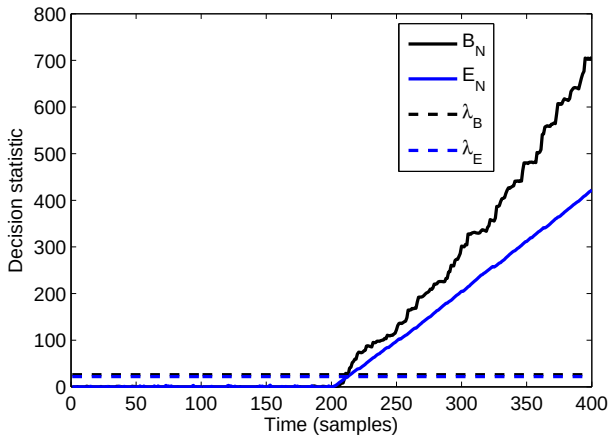

Figure 3: B and E for 400 samples with the distribution change occurring at the 200th sample.

5.1. GLRT algorithm

Figure 3 shows the simulation results for B_N and E_N for 400 samples and for σ_x^2 that lies in the range of $[0.5\sigma_{x_{avg}}^2, 2\sigma_{x_{avg}}^2]$. For the detection of empty spectrum slots, i.e., Phase I, the PU's signal leaves the spectrum at the 200th sample. For the detection of the entrance of the PU's signal, i.e., Phase II, the PU's signal enters the spectrum at the 200th sample. One can see that once the status of the spectrum changes, i.e., the distribution of the collected samples changes, B and E rapidly increases in both cases. The decision statistic is compared to a threshold (λ_B and λ_E, respectively) to decide whether the current status has changed or not.

5.2. Impact of α on seed generation

We start by presenting the impact of α on seed generation. Figure 4 shows the simulation results obtained for B_α at the legitimate SUs, when $\alpha = 2.5$ (top) and 5 (bottom), respectively. In both subfigures, the SNR is set to 15 dB at the first legitimate SU, to 10 dB at the second, and 10 dB at the malicious node. Since the malicious node does not know the value of α, it is assumed that it uses $\alpha = 1$. Although, the malicious node uses the same spectrum sensing technique, its decision statistic before the entrance of the PU's signal is almost zero making it unsuitable for secret key generation. The change in SNR between the two legitimate SUs leads to different values of B_α after the entrance of the PU's signal, exhibiting higher values as the SNR

increases. Nevertheless, the seed S, which is zoomed-in in both subfigures, is not affected by the different values of SNR, since it is generated from the samples collected before the entrance of the PU's signal. Rather, it is affected only by the value of α. As α grows, the drift in the first 200 samples of B increases. Moreover, S at both legitimate SUs is very similar. The samples used as seed at the malicious node are close to zero, making them unsuitable for link key generation. Furthermore, one can see that B_α can be used also to detect the entrance of the PU's signal instead of B, by properly adjusting the value of the threshold to account for the effect of α.

5.2.1. Seed shuffling

In Figure 5, we present a shuffled version of the samples of S. The shuffled indices set is provided in the *primary* secret to the legitimate SUs. One can see that S does not follow the continuously increasing pattern anymore. Rather, it is completely randomized.

5.3. BMR

Next, Figure 6 shows the simulation results for the BMR of the link key extracted at two legitimate SUs vs. the difference in SNR between the SUs. The SNR at SU1 varies between 0 and 20 dB, while the SNR at SU2 is fixed at 10 dB. We set $\alpha = 10$ and use different numbers of quantization bits, namely, $N_q = 4, 6$ and 8. We compare the results of our proposed algorithm to conventional channel based secret key generation algorithm [27]. Each BMR value is estimated through extensive Monte Carlo simulation using 10,000 iterations. The results clearly show that the change in SNR between the two SUs does not affect the performance of our link key generation algorithm. As expected, as N_q increases, the BMR increases, however, the achieved BMR after encoding is less than 10%. The achieved BMR before information reconciliation and privacy amplification is well below the value provided in [26] of 15%, thus leading to very good performance. The BMR achieved through our algorithm shows comparable results to channel based physical layer security scheme. However, unlike our proposed algorithm, the effect of change in SNR is clear in channel based key generation algorithm. Furthermore, it is important to stress that changing the value of α does not have much effect on the achieved BMR. BMR results presented in Figure 7 highlight that the achieved BMR for α varying between 5 and 30 is almost constant and equal to 44% before encoding, and to 11% after encoding.

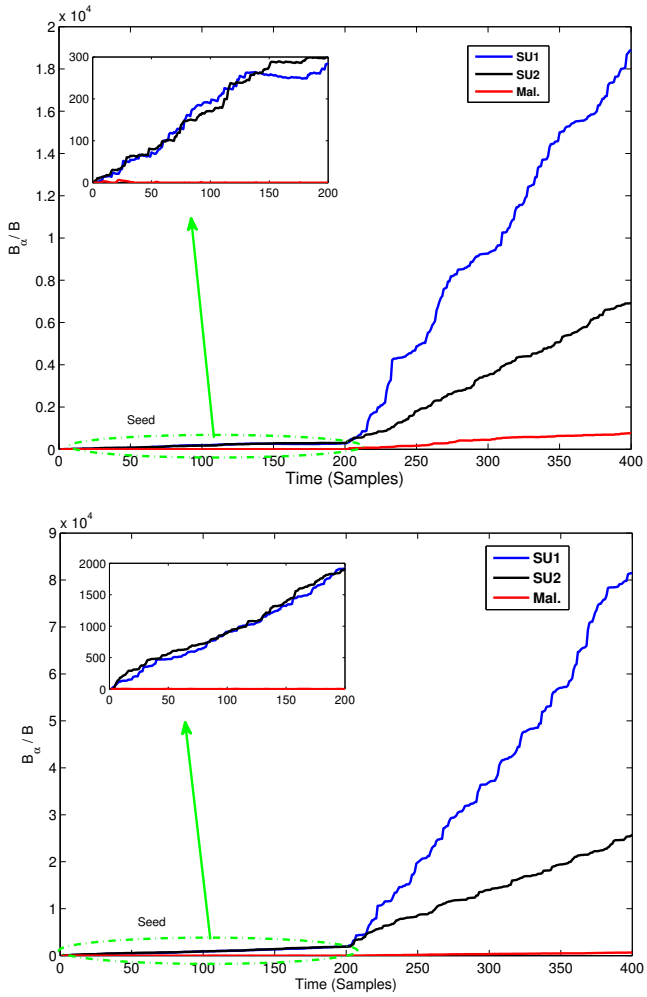

Figure 4: B_α for $\alpha = 2.5$ (top) and $\alpha = 5$ (bottom) at two legitimate SUs, and B at the malicious user. B_α and B are plotted as functions of time (400 samples, the seed is zoomed in).

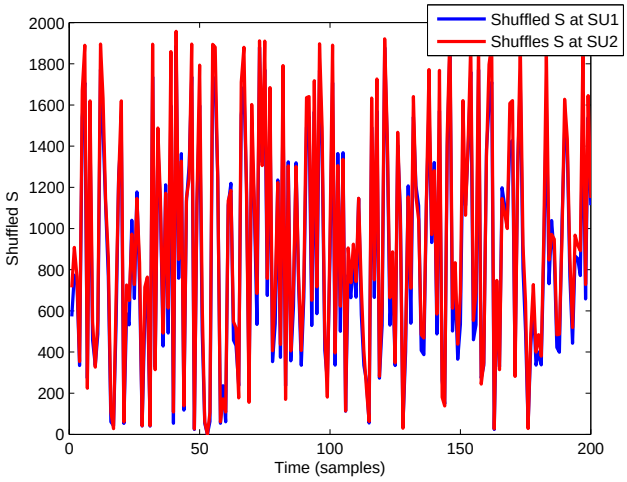

Figure 5: Shuffled S at the two legitimate SUs.

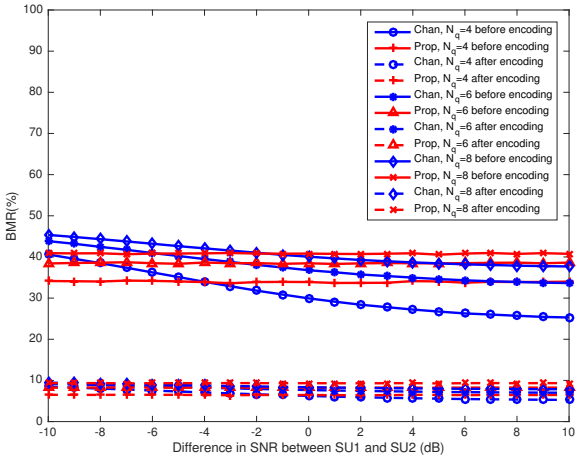

Figure 6: BMR vs. the difference in SNR between the two legitimate SUs, for different numbers of quantization bits.

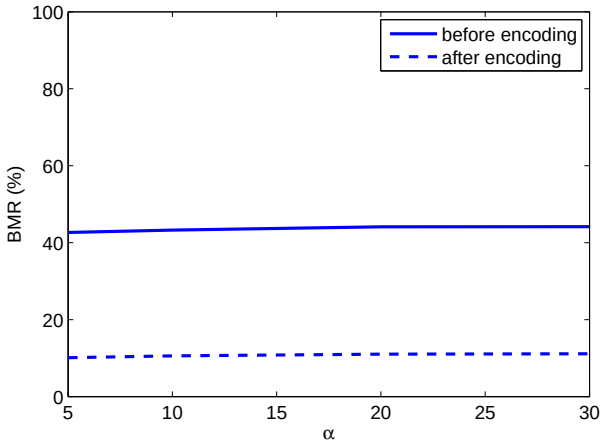

Figure 7: The BMR vs. α before and after encoding for our proposed algorithm and channel based algorithm.

5.4. Entropy

Entropy is a measure of the level of randomness of the generated key. We compare the entropy of the link key generated through our algorithm to channel based algorithm [27] in Fig. 8 for $N_q = 6$ bits. The entropy rate achieved through our proposed algorithm is comparable to that achieved by conventional channel based technique.

5.5. α vs. number of detection cycles

The way α evolves over time is depicted in Figure 9 (top), for $\alpha = 2$, $\beta = 18$, $\gamma = 5$, $\rho = 200$ and the SNRs at the two legitimate equal to 15 dB and 10 dB, respectively. One can see that using the LGN makes the estimated α to fluctuate randomly, which is exactly what we want in order to generate efficient link keys. Also, the results in Figure 9 (bottom) (obtained under the same settings) confirm that the RMSE of α is very low.

5.6. How often should the primary key be distributed?

As stated earlier, a new primary key may need to be distributed when its effect on the link key generation, through the parameter α, tends to dissolve. Then a reasonable concern is about when a new primary key should be generated. Figure 10 depicts the estimated α versus the number of detection cycles, for initial $\alpha = 2$, $\beta = 8$, $\gamma = 5$ and $\rho = 200$ (top) and $\alpha = 2$, $\beta = 30$,

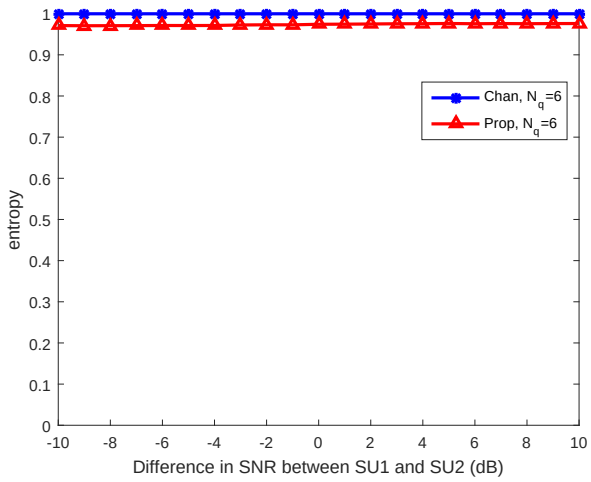

Figure 8: Entropy rate of the generated key for our proposed algorithm and channel based algorithm.

$\gamma = 10$ and $\rho = 200$ (bottom). From Figure 9 (top) and Figure 10, it can be inferred that the periodicity and randomness of the newly generated α depends on the selection of the parameters[3] β, γ, ρ and initial α. Therefore, a new primary key is distributed whenever the value of α follows a periodic pattern or does not fluctuate randomly from one detection cycle to the next as desired.

5.7. A qualitative comparison

To counteract SSDF, reputation based techniques such as the ones presented in [8, 28] require long time, i.e., many detection cycles, to build up a good reputation. In addition, reputation is built based on the overall decision, which may be incorrect in case of SSDF attacks launched by many malicious nodes. Non-reputation based techniques such as [29, 30, 31], are also based on the assumption that the overall decision is correct. On the other hand, our algorithm neither requires many detection cycles to efficiently operate, nor it assumes the correctness of the overall decision.

Typical physical layer security techniques, such as [10], used to counteract eavesdropping require extensive channel probing to generate a suitable link

[3]Refer to [22] for more details on the selection and limitations of the LGN parameters.

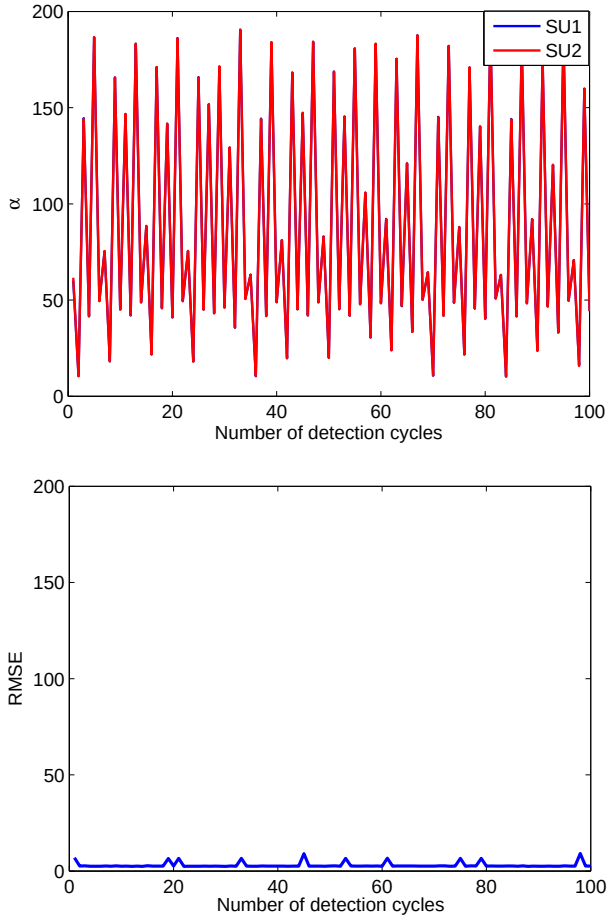

Figure 9: α_n at the two legitimate SUs as a function of the number of detection cycles (top). RMSE of the estimated α_n at the two legitimate SUs (bottom).

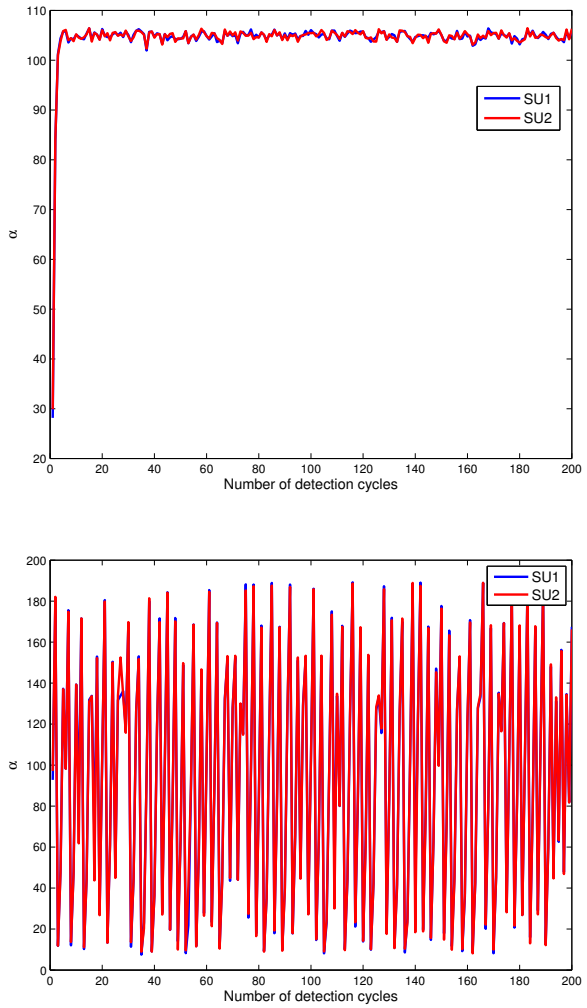

Figure 10: α_n at the two legitimate SUs as a function of the number of detection cycles: for initial $\alpha = 2$, $\beta = 8$, $\gamma = 5$ and $\rho = 200$ (top) and $\beta = 30$, $\gamma = 10$ and $\rho = 200$ (bottom).

key. The frequent channel probing requires multiple beacon exchange, synchronization and employment of a channel estimation technique. In addition, they may need an initial agreement on some parameters [10] as in our proposed technique. On the contrary, our solution exploits the spectrum sensing data, which is already available at the two legitimate nodes to extract the link key to make key exchange less frequent. Thus, our algorithm requires a shorter time to generate the link key as well as much lower computational complexity stemming from not deploying channel estimation techniques.

6. Conclusion

We focused on a cognitive radio network where symmetric key encryption is used to counteract two common physical layer attacks, namely, spectrum sensing data falsification and eavesdropping. In this context, we proposed a novel algorithm that greatly reduces the complexity of link key management. Our algorithm uses the information provided in a pre-distributed primary key to manipulate the spectrum sensing decision statistic at legitimate SUs so as to yield a seed that can be exploited for generating a new link key at each detection cycle. As a result, the solution we propose does require neither extensive processing nor exchange of beacons or synchronization data. Furthermore, the manipulated samples do not disrupt the decision statistic, rather they can be used for signal detection as well. Numerical results show that the achieved bit mismatch rate is less than 10% right after the quantization and encoding step, which is an excellent result. Also, the difference between the SNR at two legitimate SUs does not affect the performance of link key generation suggesting that our solution is suitable for practical scenarios.

[1] H. V. Poor, O. Hadjiliadis, Quickest Detection, Cambridge University Press, 2008.
URL http://dx.doi.org/10.1017/CBO9780511754678

[2] L. Lai, Y. Fan, H. Poor, Quickest detection in cognitive radio: A sequential change detection framework, in: Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, 2008, pp. 1–5. doi:10.1109/GLOCOM.2008.ECP.567.

[3] G. Baldini, T. Sturman, A. Biswas, R. Leschhorn, G. Godor, M. Street, Security aspects in software defined radio and cognitive radio networks:

A survey and a way ahead, Communications Surveys Tutorials, IEEE 14 (2) (2012) 355–379. doi:10.1109/SURV.2011.032511.00097.

[4] Z. Shu, Y. Qian, S. Ci, On physical layer security for cognitive radio networks, Network, IEEE 27 (3) (2013) 28–33. doi:10.1109/MNET.2013.6523805.

[5] A. Fragkiadakis, E. Tragos, I. Askoxylakis, A survey on security threats and detection techniques in cognitive radio networks, Communications Surveys Tutorials, IEEE 15 (1) (2013) 428–445. doi:10.1109/SURV.2011.122211.00162.

[6] A. Ghasemi, E. Sousa, Collaborative spectrum sensing for opportunistic access in fading environments, in: New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on, 2005, pp. 131 –136. doi:10.1109/DYSPAN.2005.1542627.

[7] Z. Quan, S. Cui, H. Poor, A. Sayed, Collaborative wideband sensing for cognitive radios, Signal Processing Magazine, IEEE 25 (6) (2008) 60 –73. doi:10.1109/MSP.2008.929296.

[8] R. Chen, J.-M. Park, Y. Hou, J. Reed, Toward secure distributed spectrum sensing in cognitive radio networks, Communications Magazine, IEEE 46 (4) (2008) 50–55. doi:10.1109/MCOM.2008.4481340.

[9] C. da Silva, B. Choi, K. Kim, Distributed spectrum sensing for cognitive radio systems, in: Information Theory and Applications Workshop, 2007, 2007, pp. 120–123. doi:10.1109/ITA.2007.4357570.

[10] S. Mathur, W. Trappe, N. Mandayam, C. Ye, A. Reznik, Radiotelepathy: Extracting a secret key from an unauthenticated wireless channel, in: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, MobiCom '08, ACM, New York, NY, USA, 2008, pp. 128–139. doi:10.1145/1409944.1409960.
URL http://doi.acm.org/10.1145/1409944.1409960

[11] Y. Liu, L. Wang, T. T. Duy, M. Elkashlan, T. Q. Duong, Relay selection for security enhancement in cognitive relay networks, IEEE Wireless Communications Letters 4 (1) (2015) 46–49. doi:10.1109/LWC.2014.2365808.

[12] L. Fan, S. Zhang, T. Q. Duong, G. K. Karagiannidis, Secure switch-and-stay combining (sssc) for cognitive relay networks, IEEE Transactions on Communications 64 (1) (2016) 70–82. doi:10.1109/TCOMM.2015.2497308.

[13] L. Zhang, R. Zhang, Y.-C. Liang, Y. Xin, S. Cui, On the relationship between the multi-antenna secrecy communications and cognitive radio communications, Communications, IEEE Transactions on 58 (6) (2010) 1877–1886. doi:10.1109/TCOMM.2010.06.090063.

[14] S. Vaudenay, A Classical Introduction to Cryptography: Applications for Communications Security, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[15] W. Fumy, P. Landrock, Principles of key management, IEEE Journal on Selected Areas in Communications 11 (5) (1993) 785–793. doi:10.1109/49.223881.

[16] M. M. Prabhu, S. Raghavan, Security in computer networks and distributed systems, Computer Communications 19 (5) (1996) 379 – 388. doi:http://dx.doi.org/10.1016/0140-3664(95)01031-9.
URL http://www.sciencedirect.com/science/article/pii/0140366495010319

[17] P. Halliden, Network security issues, Computer Communications 13 (10) (1990) 626 – 629. doi:http://dx.doi.org/10.1016/0140-3664(90)90090-4.
URL http://www.sciencedirect.com/science/article/pii/0140366490900904

[18] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, N. B. Mandayam, Information-theoretically secret key generation for fading wireless channels, Information Forensics and Security, IEEE Transactions on 5 (2) (2010) 240–254. doi:10.1109/TIFS.2010.2043187.

[19] S. A. Camtepe, B. Yener, Key distribution mechanisms for wireless sensor networks: a survey, Rensselaer Polytechnic Institute, Troy, New York, Technical Report (2005) 05–07.

[20] Y. C. Liang, Y. Zeng, E. C. Y. Peh, A. T. Hoang, Sensing-throughput tradeoff for cognitive radio networks, IEEE Transactions on Wireless

Communications 7 (4) (2008) 1326–1337. doi:10.1109/TWC.2008.060869.

[21] L. Lu, G. Y. Li, S. Li, Optimum periodic spectrum sensing for cr networks, IEEE Communications Letters 16 (12) (2012) 1–4. doi:10.1109/LCOMM.2012.111612.121162.

[22] S. Tezuka, Uniform Random Numbers Theory and Practice, Vol. 315, 1st Edition, Springer Science+Business Media,LLC.

[23] L. Tan, Digital Signal Processing Fundamentals and Applications, Academic Press, 2007.

[24] J. Zhang, S. Kasera, N. Patwari, Mobility assisted secret key generation using wireless link signatures, in: INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–5. doi:10.1109/INFCOM.2010.5462231.

[25] G. Brassard, L. Salvail, Secret-key reconciliation by public discussion, Springer-Verlag, 1994, pp. 410–423.

[26] O. Gungor, F. Chen, C. Koksal, Secret key generation via localization and mobility, Vehicular Technology, IEEE Transactions on PP (99) (2014) 1–1. doi:10.1109/TVT.2014.2342714.

[27] A. Sayeed, A. Perrig, Secure wireless communications: Secret keys through multipath, in: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, 2008, pp. 3013–3016. doi:10.1109/ICASSP.2008.4518284.

[28] P. Kaligineedi, M. Khabbazian, V. Bhargava, Malicious user detection in a cognitive radio cooperative sensing system, Wireless Communications, IEEE Transactions on 9 (8) (2010) 2488–2497. doi:10.1109/TWC.2010.061510.090395.

[29] F. Adelantado, C. Verikoukis, A non-parametric statistical approach for malicious users detection in cognitive wireless ad-hoc networks, in: Communications (ICC), 2011 IEEE International Conference on, 2011, pp. 1–5. doi:10.1109/icc.2011.5963004.

[30] E. Soltanmohammadi, M. Naraghi-Pour, Fast detection of malicious behavior in cooperative spectrum sensing, Selected Areas in Communications, IEEE Journal on 32 (3) (2014) 377–386. doi:10.1109/JSAC.2014.140301.

[31] W. Wang, L. Chen, K. Shin, L. Duan, Thwarting intelligent malicious behaviors in cooperative spectrum sensing, Mobile Computing, IEEE Transactions on 14 (11) (2015) 2392–2405. doi:10.1109/TMC.2015.2398446.